**To my parents,
on their 50th anniversary**

Preface to Volume II

If it is bad luck to title a book “Volume I”, as Gian-Carlo Rota says in *Indiscreet Thoughts*, it is not good luck to promise a Volume II in the introduction, either. For the ten years after the appearance of *Classical Recursion Theory* in 1989, I was torn between the easy choice of publishing a new edition of the first volume with an amended introduction, and the much harder choice of completing the second volume. Now that an apparently diverging sequence of successive drafts has finally come to a limit, I can release both a new edition of the first volume and a second volume.

In this last moment dedicated to thanksgivings, my first thoughts go to Laura. She entered my shattered life in 1986, putting its pieces back together and providing an inexhaustible flow of joy, happiness and understanding. She whipped me back into line, whenever I strayed from what she rightly considered the correct path. As I already wrote on her copy of Volume I, this book is also hers.

As the previous one, much of my last decade has been enlightened by visits to different parts of the world, made possible by a number of friends. First and foremost, Anil Nerode and Richard Shore, thanks to whom Ithaca became my second home for three years and thirteen summers. Then John Crossley in Melbourne in 1988, Dongping Yang in Beijing in 1992 and 1995, Gerald Sacks in Boston in 1995, 1996 and 1998, Antonin Kučera in Prague and Cristian Calude in Auckland in 1996, and Ding Decheng in Nanjing in 1998. Last but not least, Andrea Sorbi, whose selfless work as coordinator of a Human Capital and Mobility Project provided the funds for a number of European trips.

Even more than for the first volume, I owe a great debt to the colleagues who have read parts of the manuscript and have provided corrections and suggestions: Klaus Ambos-Spies, Francesco Bergadano, Cristian Calude, Barry Cooper, Ugo de' Liguoro, Lavinia Egidi, Dick Epstein, Matt Giorgi, Lane Hemachandra, Carl Jockusch, Martin Kummer, Antonin Kučera, Steffen Lempp, Bob Lubarsky, Wolfgang Merkle, Franco Montagna, Michael Morley, Dan Osherson, Alan Selman, Mark Simpson, Ted Slaman, Bob Soare, Andrea Sorbi, Frank Stephan, Helmut Veith and Yue Yang.

Whatever vision informs the book has been inspired by a handful of people, whose thoughts have provided and sustained inspiration through the years: Barry Cooper, Juris Hartmanis, Carl Jockusch, Georg Kreisel, Anil Nerode, Richard Platek and Gerald Sacks. Their acquaintance has been an undeserved honor, their teaching a much appreciated gift.

However, the book would never have been completed without the massive help of the Magnificent Four whose expectations have set my standards: Rod Downey, Bill Gasarch, André Nies and Richard Shore. As friends, they have provided a constant stimulus and encouragement, much needed in the face of hurdles and doubts. As colleagues, they have dedicated an enormous amount of time and energy to help me with explanations and proofs. If this volume does not displease them, I will be delighted.

As Beaumarchais once noticed, books are for authors as babies are for mothers: conceived in pleasure, carried with fatigue and given birth in pain. No words could better describe an enterprise that literally took away half of my life, nor better introduce the subject of dedication. Because, if I look back at my forty-eight years, in them I see only my parents more constant than this book. As a first child, I shared most of their life together: perhaps not as close and near as they would have liked, perhaps closer and nearer than they might have guessed. The deadline of their fiftieth anniversary on September 17th, 1999 has provided a major drive towards the completion of the book: another item to add to a long list of valuable parental offerings, which neither spoken nor printed words are able to match.

Ithaca - Torino
1989 - 1999

Contents

Introduction to Volume II

We obviously keep in Volume II the same notation and conventions used in Volume I. For the reader's convenience, we reproduce here the parts of the Introduction to Volume I which are relevant to Volume II.

What is in the Book

Recall that **Classical Recursion Theory is the study of real numbers or, equivalently, functions over the natural numbers**. The basic methods of analysis of the real numbers used in Volumes I and II are:

Hierarchies. A hierarchy is a stratification of a class of reals built from below, starting from a subclass that is taken as primitive (either because well understood, or because already previously analyzed), and obtained by iteration of an operation of class construction.

Degrees. Degrees are equivalence classes of reals under given equivalence relations, that identify reals with similar properties. Once a class of reals has been studied and understood, degrees are usually defined by identifying reals that look the same from that class point of view.

As might be imagined the two methods are complementary: first a class is analyzed in terms of intrinsic properties, for example by appropriately stratifying it in hierarchies, and then the whole structure of real numbers is studied modulo that analysis with the appropriate notion of degrees induced by the given class.

The previous complementarity is exploited throughout Volume II. In Chapter VIII we provide a study of **polynomial time computable functions** and of the induced notion of **polynomial time degrees** (similar notions of degrees could be introduced for most complexity classes studied in Chapter VIII). In Chapters IX, X and XIV we provide a study of the **recursively enumerable sets** and of the induced notion of **enumeration degrees**, while in Chapters

XII and XIII we provide a study of the **arithmetical sets** and of the induced notion of **arithmetical degrees**.

We now outline the skeleton of Volume II in more detail, referring to the introductions of the various chapters for more details. Chapters VII and VIII resume the analysis of the fundamental objects in Recursion Theory, the recursive sets and functions, and provide a microscopic picture of them. We start in Chapter VII with an abstract study of the complexity of computation of recursive functions. Then in Chapter VIII we attempt to build from below the world of recursive sets and functions that was previously introduced in just one go. A number of subclasses of interest from a computational point of view are introduced and discussed, among them: the **polynomial time (or space) computable functions** which provide an upper bound for the class of feasibly computable functions (as opposed to the abstractly computable ones); the **elementary functions**, which are the smallest known class of functions closed under time (deterministic or not) and space computations; the **primitive recursive functions**, which are those computable by the 'for' instruction of programming languages like PASCAL, i.e. with a preassigned number of iterations (as opposed to the recursive functions, computable by the 'while' instruction, which permits an unlimited number of iterations); the **ϵ_0-recursive functions**, which are those provably total in Peano Arithmetic.

Chapters IX and X return to the treatment of recursively enumerable sets. A good deal of information on their structure was already gathered in Chapter III, but here a systematic study of the structures of both the lattice of **recursively enumerable sets** and of the partial ordering of **recursively enumerable degrees** is undertaken. Special tools for their treatment are introduced, most prominent among them being the **priority method**, a constructive variation of the Baire Category method.

Chapter XI deals with **limit sets**, also known as Δ_2^0 sets, which are limits of recursive functions. They are a natural formalization of the notion of sets for which membership can be determined by effective trials and errors, unlike recursive sets (for which membership can be effectively determined), and recursively enumerable sets (for which membership can be determined with at most one mistake, by first guessing that an element is not in the set, and then changing opinion if it shows up during the generation of the set).

Chapter XII deals with **arithmetical sets**, which are definable in the language of First-Order Arithmetic. As a special tool for their treatment we introduce the method of **arithmetical forcing**, which can be combined with the Baire Category and the priority methods. Chapter XIII studies the structure of the continuum w.r.t. the notion of relative arithmetical definability provided by the **arithmetical degrees**, along the lines of Chapter V. Similarly, Chapter XIV studies the structure of the continuum w.r.t. a notion of relative recursive enumerability provided by the **enumeration degrees**.

Starred subsections deal with topics related to the ones at hand thought sometimes quite far away from the immediate concern. They provide those connections of Recursion Theory to the rest of mathematics and computer science which make our subject part of a more articulate and vast scientific experience. Limitations of our knowledge and expertise in these fields make our treatment of the connections rather limited, but we feel they add important motivation and direct the reader to more detailed references. In particular, we continue the commentary made throughout the book on the relationships with **computers**, **logic**, and the theory of **formal systems**.

As already in Volume I, we have opted for breadth rather than depth and have provided rudiments of many branches of Classical Recursion Theory, rather than complete and detailed expositions of a small number of topics. In this respect our book is in the tradition of Kleene [1952] and Rogers [1967], and differs from texts such as Lerman [1983], Rose [1984], Soare [1987], Balcázar, Díaz and Gabarró [1988], [1990], and Papadimitriou [1994], which can be used as useful complements and advanced textbooks in their specialized areas (see also the relevant chapters of the *Handbook of Recursion Theory*, in particular Ambos-Spies [1999], Chong and Friedman [1999], Cooper [1999], Odifreddi [1999], Schwichtenberg [1999], Shore [1999], Slaman [1999] and Soare [1999]).

How to Use the Book

This book has been written with two opposite, and somewhat irreconcilable, goals: to provide for both an adequate textbook, and a reference manual. Supposedly, the audiences in the two cases are different, consisting mainly of students in the former, and researchers in the latter. This has resulted in different styles of exposition, reflecting different primary goals: self-containment and detailed explanations for textbooks, and completeness of treatment for manuals. We have tried to solve the dilemma by giving a detailed treatment of the main topics in the text, and sketches of the remaining arguments in the exercises and in the starred parts.

The **exercises** usually cover material directly connected to the subject just treated and provide hints of proofs in the majority of cases, in various degrees of detail. In a few cases, for completeness of treatment and easiness of reference, some of the exercises use notions or methods of proof introduced later in the book.

The **starred chapters and sections** treat topics that can be omitted on a first reading. The **starred subsections** deal with side material, usually giving broad overviews of subjects that are more or less related to the main flow of thought, but which we believe provide interesting connections of Recursion Theory with other branches of Logic or Mathematics. The style is mostly suggestive: we try to convey the spirit of the subject by quoting the main results and, sometimes, the general ideas of their proofs. Detailed references are usually given, both for the original sources and for appropriate updated treatments.

The general prerequisite for Volume II is a working knowledge of the rudiments of Recursion Theory. When dealing with applications, knowledge of the subject will be assumed but, since the treatment is kept separate from the main text, there will be no loss in skipping the relative parts.

The chapters have been kept self-contained as far as possible. We have done our best to keep the style informal and devoid of technicalities, and we have resorted to technical details only when we have not been able to avoid them, no doubt because of our inadequacy.

Instead of the usual complicated diagrams of dependencies, we give suggestions on how the two volumes of the book can be used as a textbook for classes in which Recursion Theory is the main ingredient.

Complexity Theory

Chapters VII and VIII deal with abstract complexity theory and complexity classes, and do not require any background, except for a working knowledge of recursiveness and Turing machines (like Sections 1 and 4 of Chapter I). The treatment is fairly complete but, going beyond the usual unbalanced confinement to polynomial time and space computable functions, it also covers unjustly neglected classes of recursive functions, such as elementary, primitive recursive, and ϵ_0-recursive functions which are of interest to the computer scientist.

Recursively Enumerable Sets

The elementary theory of r.e. sets and degrees is contained in Chapter III, which requires only some background in elementary Recursion Theory. The chapter goes up to the solution to Post's problem (Sections 1 to 5) and the basic classes of r.e. sets. It can be used either as a final section of a course on elementary Recursion Theory (not dealing with alternative definitions of recursiveness), or as the initial segment of an advanced course on r.e. sets. In the latter case, it should be followed by Chapter IX, dealing with the lattice of r.e. sets, and a choice of material from Chapter X, in which priority arguments are introduced. Some of the material here, e.g. the theory of r.e. m-degrees, is not standard, but is useful in various respects: intrinsically, this structure is much better behaved than the schizoid one of r.e. T-degrees, and it reflects the global structure of degrees, which the latter does not; moreover, arguments on T-degrees (such as various coding methods) are better understood in their simpler versions for m-degrees.

Degree Theory

Elementary degree theory is treated in Chapter V which, with some background in elementary Recursion Theory, can be read autonomously. We develop the theory up to a point where it is possible to prove a number of global results. This forms the nucleus of a course, and it can be followed by a number of advanced topics including a choice of results from Chapters XI and XII, on degrees of Δ_2^0 and arithmetical sets. Chapter VI, on m-degrees, is often unjustly neglected, but it does provide the only existing example of global characterization of a structure of degrees. It can be read independently of Chapter V.

Notation and Conventions

$\omega = \{0, 1, \dots\}$ is the set of natural numbers, with the usual operations of plus $(+)$ and times $(\times$ or $\cdot)$, and the order relation $\leq$. $\mathcal{P}(\omega)$ is the power set of ω, i.e. the set of all subsets of ω. ω^ω and $\mathcal{P}$ are, respectively, the sets of total and partial functions from ω to itself.

We reserve certain lower or upper case letters to denote special objects:

- $a, b, c, \dots, x, y, z, \dots$ for natural numbers
- $f, g, h, \dots$ for total functions of any number of variables
- $\alpha, \beta, \gamma, \dots, \varphi, \psi, \chi, \dots$ for partial functions of any number of variables
- $F, G, H, \dots$ for functionals, i.e. functions with some variables ranging over numbers, and some over functions
- $A, B, C, \dots, X, Y, Z, \dots$ for sets of natural numbers
- $P, Q, R, \dots$ for predicates of any number of variables
- $\sigma, \tau, \dots$ for strings, i.e. partial functions with finite domain and values in $\{0, 1\}$.

Regarding **sets**:

- $x \in A$ means that x is an element of A
- $|A|$ is the cardinality of A, i.e. the number of its elements
- $A \subseteq B$ and $A \subset B$ are the relations of inclusion and strict inclusion
- $\overline{A}$ is the complement of A, and the prefix 'co-' in front of a property of a set means that the complement has this property (i.e. a set is co-immune if its complement is immune)

- $A \cup B$ is the union of A and B, i.e. the set of elements belonging to at least one of A and B
- $A \oplus B$ is the disjoint union of A and B, i.e. the set of elements of the form $2x$ if $x \in A$, and $2x+1$ if $x \in B$
- $A \cap B$ is the intersection of A and B, i.e. the set of elements belonging to both A and B
- $A \times B$ is the cartesian product of A and B, i.e. the set of pairs (x, y) whose first and second components are, respectively, in A and B
- $A \cdot B$ is the recursive product of A and B, i.e. the set of codes $\langle x, y \rangle$ of pairs $(x, y) \in A \times B$
- c_A is the characteristic function of A, with value 1 if the given argument is in the set, and 0 otherwise.

Regarding **predicates**:

- $\neg P$, $P \wedge Q$, $P \vee Q$, $P \rightarrow Q$, $P \leftrightarrow Q$, $\forall x P$, $\exists x P$ are the usual logical operations of negation, conjunction, disjunction, implication, equivalence, universal and existential quantification.

  The symbols $\rightarrow$ and $\leftrightarrow$ will be used in a formal way, to build new properties from given ones. The symbols $\Rightarrow$ and $\Leftrightarrow$ will be used informally, as abbreviations for 'if ... then', and 'if and only if'.

  We use bounded quantifiers as abbreviations:

$$\begin{array}{lll} (\exists x \leq y)P(x) & \text{for} & (\exists x)[x \leq y \ \wedge \ P(x)] \\ (\forall x \leq y)P(x) & \text{for} & (\forall x)[x \leq y \rightarrow P(x)]. \end{array}$$

- c_P is the characteristic function of P, with value 1 if P holds for the given argument and 0 otherwise.

Regarding **binary relations** on a set A, R is:

- reflexive if xRx for every $x \in A$
- antireflexive if $\neg(xRx)$, for every $x \in A$
- symmetric if $xRy \Rightarrow yRx$ for every $x, y \in A$
- transitive if $xRy \wedge yRz \Rightarrow xRz$ for every $x, y, z \in A$
- a (weak) partial ordering if it is reflexive and transitive (weak partial orderings are indicated by $\leq$, $\preceq$, or $\sqsubseteq$)

- a (strict) partial ordering if it is antireflexive and transitive (strong partial orderings are indicated by $<$, $\prec$, or $\sqsubset$)
- a total ordering if it is a partial ordering, and $xRy \vee yRx \vee (x = y)$ for every $x, y \in A$
- an equivalence relation if it is reflexive, transitive, and symmetric; in this case the set A is partitioned into equivalence classes (each consisting of the elements that are in the relation R with each other)
- an uppersemilattice if any pair of elements of A has a l.u.b., and a lattice if any pair of elements of A has both l.u.b. and g.l.b. (given two elements x and y, their least upper bound (l.u.b.) and greatest lower bound (g.l.b.) are, respectively, the smallest element of A greater than both x and y, and the greatest element of A smaller than both x and y).

Regarding **functions**:

- $f \circ g$ or fg denote the composition of f and g
- $f^{(n)}$ denotes the result of n iterations of f, i.e. n successive applications of f (by convention, $f^{(0)}(x) = x$)
- $\varphi(x)\downarrow$ means that φ is defined on x
- $\varphi(x)\uparrow$ means that φ is undefined on x
- the set of elements on which φ is defined is called its domain, and the set of elements which are values of φ for some argument is called its range
- $\varphi \simeq \psi$ means that φ and ψ are equal as partial functions, i.e. on each argument they are either both undefined, or both defined and equal
- the set of pairs (x, y) such that $\varphi(x) \simeq y$ is called the graph of φ
- $\alpha \subseteq \beta$ means that as partial functions β extends α, i.e. if α is defined on an argument, then β is too and has the same value.

Each chapter is divided into numbered sections, and each section is divided into unnumbered subsections. There is a unique progressive numbering inside sections, including definitions, results, and exercises. Internal references in a given chapter may omit the chapter number.

The bibliography only includes papers quoted in the book. We have done our best to attribute results and quote the original sources. In case of unpublished results, when an attribution has been possible through personal communication or other sources we have attached names without references, and the mistakes that may have occurred are unintentional. We are, of course, well aware of the fact that simply quoting original sources is only a ghost of history, and it barely hints at the growth and interaction of ideas. But at least it provides the bare facts.

It is once again time to plunge into the real work. We hope you will find Volume II readable, despite the abstraction of some of its notions and the intricacies of some of its arguments. As in the second leg of another challenging but rewarding trip,

per correr migliori acque alza le vele
ormai la navicella . . .
(Dante, *Purgatorio*, I.1–2)

Chapter VII

Theories of Recursive Functions

In Chapters I and II we have introduced the class of recursive functions from a global point of view, by describing a number of distinct but equivalent formalisms to define and compute them. We now provide a local analysis aimed at a classification of the recursive functions, in terms of two different approaches.

First, we deal with the **computational complexity** of the recursive functions. In Section 1 we introduce two different types (static and dynamic) of measures for the complexity of programs. In Section 2 we consider *single total recursive functions* and discuss upper and lower bounds of the complexities of their programs, as well as the notion of best program. In Section 3 we consider *classes of total recursive functions* defined by fixed complexity bounds, develop tools to generate various partial hierarchies of the recursive functions, and prove the impossibility of obtaining a satisfactory exhaustive hierarchy. In Section 4 we turn to more concrete measures and study *time and space* constraints for Turing machine programs.

Second, we deal with the **inductive inferability** of the recursive functions. In Section 5 we introduce a number of notions formalizing the process of *learning classes of recursive functions*, ranging from a strong notion that learns any of the usual subclasses of recursive functions studied in Chapter VIII, to a weak notion that learns the class of all total recursive functions.

Since we often work with properties that hold almost everywhere, in the sense of being true for all numbers with at most finitely many exceptions, we introduce the following **notation**:

1. $\exists_\infty \boldsymbol{x}$ means 'there exist infinitely many x', and is sometimes also written

as '**i.o.**' (infinitely often)

2. $\forall_\infty \boldsymbol{x}$ means 'for all x, with at most finitely many exceptions', and is sometimes also written as '**a.e.**' (almost everywhere).

Since many results in Complexity Theory are sensitive to the details of data representation, we introduce a further notation:

3. $|\boldsymbol{x}|$ means 'the length of x', and is proportional to either x or $\log_n x$, according to whether x is represented in unary or n-ary ($n \geq 2$) notation, respectively.

VII.1 Measures of Complexity

Recursive functions were introduced in Chapter I through a series of equivalent formalisms that allow their computations in various ways.

A first classification of the recursive functions is obtained by looking at *structural restrictions* of Turing machines. For example, finite automata, push-down automata, and stack automata are all natural subclasses of Turing machines, which compute only subclasses of the recursive functions. Despite the intrinsic interest in this area we only glimpse at it and do not study it in detail, the main reason being that we are looking for fine classifications of the recursive functions, while this approach only provides a very rough one. The reader is referred to Minsky [1967], Arbib [1969], Savage [1976], Hopcroft and Ullman [1979], and Lewis and Papadimitriou [1981] for treatments.

A second classification is achieved by looking at computational devices as objects, assigning them a complexity evaluation (a number measuring their 'size'), and comparing devices by comparing their sizes. This is a *static approach*, and natural examples of measures of size are given by the numerical value of an index for a recursive function (obtained from any of the formalisms discussed in Chapter I), or by the number of symbols needed to write down a program.

A third classification is accomplished by looking at the behavior of computations on given inputs. Here a single number is not enough, and a function is needed to code such a behavior (which can vary widely on given inputs). This is a *dynamic approach*, and natural examples of measures of behavior are given by the number of moves or scanned cells used by a Turing machine in a computation (as functions of the inputs).

We concentrate in this chapter on the dynamic approach, but first dispose briefly of the static approach, since it provides a flavor of the typical results for dynamic measures, in a simpler setting.

Static complexity measures $\star$

As a first approximation, we introduce static complexity measures in an abstract way as follows: given an acceptable system of indices $\{\varphi_e\}_{e\in\omega}$ for the partial recursive functions (see II.5.2), we call a *static complexity measure* any total recursive function m, and call *complexity* or *size of* e the number $m(e)$. Then we can naturally talk of the *minimal complexity* of a partial recursive function as the least complexity of its programs (this notion obviously depends on both m and the acceptable system).

Exercises VII.1.1 (Kloss [1964]) a) *m takes its minimum value finitely often if and only if, for any acceptable system of indices, there are functions with different minimal complexity (relative to m).* (Hint: let x be the minimum value assumed by m and define

$$e \in A \ \Leftrightarrow\ m(e) = x.$$

A is recursive. If A is infinite, let f be a one-one recursive function with range A, and

$$\widehat{\varphi}_e \simeq \begin{cases} \varphi_{f^{-1}(e)} & \text{if } e \in A \\ \varphi_0 & \text{otherwise.} \end{cases}$$

Then $\{\widehat{\varphi}_e\}_{e\in\omega}$ is an acceptable system and every partial recursive function has an index belonging to A. In particular, x is the minimal complexity of every recursive function.)

b) *m takes each of its values finitely often if and only if, for any acceptable system of indices, there are infinitely many functions with pairwise different minimal complexity (relative to m).* (Hint: similar to part a), by considering the first value that m takes infinitely often.)

Motivated by the exercises above, we modify our first definition of static complexity measure as follows.

Definition VII.1.2 (Blum [1967a]) *Given an acceptable system $\{\varphi_e\}_{e\in\omega}$ for the partial recursive functions, a* **static complexity measure** *is a recursive function m such that the function*

$$x \longmapsto \textit{number of elements of } \{e : m(e) = x\}$$

is recursive.

$m(e)$ is variously referred to as the **static measure**, the **static complexity**, the **size**, or the **length** of e.

Actually, knowing the number of elements of $\{e : m(e) = x\}$ allows us to find them recursively in x (by computing successive values of m, until the right number of e's such that $m(e) = x$ has been found). Thus we can say that m is a static complexity measure if it gives the same size only to finitely many programs, and we know exactly which programs have a given size. The first condition avoids a collapse of the notion of minimal complexity (see VII.1.1.b), while the second is used in the proof of VII.1.3.

The most typical example of a static measure is the *length of a program*, which can be measured either directly, e.g. by counting the number of symbols or of instructions, or indirectly, e.g. by considering the size of the index coding the program (thus taking advantage of the fact that an index is a number).

Proposition VII.1.3 Recursive Relatedness of Static Measures (Blum [1967a]) *Given any two static measures m_1 and m_2, there is a recursive function g such that, for every e,*

$$m_1(e) \leq g(m_2(e)) \qquad \textit{and} \qquad m_2(e) \leq g(m_1(e)).$$

Proof. To obtain $g(m_2(e)) \geq m_1(e)$ it is enough to let

$$g(x) \geq m_1(e), \text{ for any } e \text{ such that } m_2(e) = x.$$

Similarly, to obtain $g(m_1(e)) \geq m_2(e)$ it is enough to let

$$g(x) \geq m_2(e), \text{ for any } e \text{ such that } m_1(e) = x.$$

We thus let

$$g(x) = \max_{e \in I_x} \{m_1(e), m_2(e)\},$$

where

$$I_x = \{e : m_1(e) = x \text{ or } m_2(e) = x\}.$$

By Definition VII.1.2, I_x is finite and can be obtained recursively from x. Then g is recursive. □

The next result belongs to a family of so-called *speed-up theorems*, whose underlying abstract feature is that in certain situations one can 'do better'. A number of other examples are given in this chapter (see VII.1.11, VII.2.16, VII.4.8, and VII.4.9).

Theorem VII.1.4 Size Shrinkage Theorem (Blum [1967a]) *Let m be a static measure. Given an infinite r.e. set of programs B and a recursive function h, we can find programs e and i computing the same function, such that $e \in B$ and*

$$h(m(i)) \leq m(e).$$

In other words, at least some function φ_e with $e \in B$ has programs much smaller in size than e.

Proof. Let

$$f(i) = \mu y\,[y \text{ is generated in } B \ \wedge\ h(m(i)) \le m(y)].$$

Since B is r.e., f is recursive. Moreover, since B is infinite and m takes each of its values only finitely often, f is total.

By the Fixed-Point Theorem (II.2.10), there exists i such that $\varphi_i \simeq \varphi_{f(i)}$. If $e = f(i)$, then:

- $\varphi_i \simeq \varphi_{f(i)} \simeq \varphi_e$
- $e = f(i) \in B$ (by definition of f)
- $h(m(i)) \le m(f(i)) = m(e)$. $\square$

Minimal or best programs (relative to a given static measure) always exist for a given function (and there may be more than one, since m is in general many-one). Precisely, the set of *minimal programs* for partial recursive functions is defined as follows:

$$M = \{e : (\forall i)[\varphi_e \simeq \varphi_i \ \Rightarrow\ m(e) \le m(i)]\}$$

The next result shows that minimal programs are difficult to generate.

Corollary VII.1.5 *M is immune.*

Proof. If M is not immune, let B be an infinite r.e. subset of it and let h be the successor function. Then the Size Shrinkage Theorem gives e and i such that

$$m(i) < m(e) \ \wedge\ e \in M \ \wedge\ \varphi_i \simeq \varphi_e,$$

contradicting the definition of M. $\square$

The previous result implies that, for example, we cannot generate an infinite r.e. sequence of minimal programs for constant functions.

Exercises VII.1.6 Minimal indices of the recursive functions. In the following exercises we consider the special case of M relative to the measure $m(e) = e$.

a) *M is not hyperimmune.* (Meyer [1972]) (Hint: let h be a recursive function such that $h(n)$ is an index for the constant function with value n. If

$$f(n) = \max\{h(x) : x < n\},$$

below $f(n)$ there are at least n minimal indices.)

b) *M is $\Sigma^0_2 - \Pi^0_2$, but is not Σ^0_2-complete.* (Meyer [1972]) (Hint: M is not Σ^0_2-complete because it is immune. $M \in \Sigma^0_2$, since

$$e \in M \ \Leftrightarrow\ (\forall i)_{i<e}(\varphi_e \not\simeq \varphi_i).$$

To show that $M \notin \Pi^0_2$, consider the Π^0_2-complete set defined by

$$e \in A \ \Leftrightarrow\ \varphi_e \text{ is the constant function with value } 0.$$

Let e_0 be the least element of A. Then

$$e \in A \ \Leftrightarrow\ e \geq e_0 \ \wedge\ (\forall i \leq e)(i \in M \wedge i \neq e_0 \Rightarrow \varphi_i \not\simeq \varphi_e).$$

If $M \in \Pi^0_2$, then $A \in \Sigma^0_2$, which is a contradiction.)

c) *M has degree* $\mathbf{0}''$. (Meyer [1972]) (Hint: let A be as in part b). Then A is r.e. in $M \oplus \mathcal{K}$, and $\overline{A}$ is r.e. in $\mathcal{K}$ by definition; thus $A \leq_T M \oplus \mathcal{K}$. Since A is Π^0_2-complete and $M \in \Sigma^0_2$, it is enough to show $\mathcal{K} \leq_T M$. Let e_1 be the minimal index of the completely undefined function, and

$$g(n) = \max\{\varphi_i(x_i) : i \leq n \wedge i \neq e_1 \wedge i \in M\},$$

where x_i is the first element generated in $\mathcal{W}_i$, which exists since $i \in M$ and $i \neq e_1$. Then $g \leq_T M$. Note that $g(n)$ is at least as big as the value of any constant function with an index $\leq n$. Let

$$\varphi_{h(z)}(x) \simeq \begin{cases} \mu s\,(z \in \mathcal{K}_s) & \text{if } z \in \mathcal{K} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Then

$$z \in \mathcal{K} \ \Leftrightarrow\ \varphi_{h(z)}(0)\!\downarrow \wedge\, z \in \mathcal{K}_{\varphi_{h(z)}(0)} \ \Leftrightarrow\ z \in \mathcal{K}_{g(h(z))}.$$

Thus $\mathcal{K} \leq_T M$.)

d) $M \equiv_T \{\langle i, e\rangle : \varphi_i \simeq \varphi_e\}$. *In particular, if we know how to recognize minimal programs, then we also know how to tell whether two programs compute the same functions.* (Constable, Drumm, Meyer) (Hint: given two indices e and i we find, recursively in M, e_1 and i_1 minimal such that $\varphi_{e_1} \simeq \varphi_e$ and $\varphi_{i_1} \simeq \varphi_i$: then

$$\varphi_e \simeq \varphi_i \ \Leftrightarrow\ e_1 = i_1.$$

To do this, given an index, we first see if it is in M. If not, then there is exactly one minimal index below it. For all the others, there must be an argument on which they disagree with the given index. Recursively in $\mathcal{K} \leq_T M$ we can eliminate all candidates, until only one remains.)

Parts b) and c) show that $M \not\equiv_m \emptyset''$ and $M \equiv_T \emptyset''$, respectively. Fenner and Schaefer [1999] have proved that $M \not\equiv_{btt} \emptyset''$. Kinber [1977] has proved that there is an acceptable system of indices such that $M \equiv_{tt} \emptyset''$, but it is not known whether this holds for every acceptable system of indices. For more on M, see Schaefer [1998].

After studying 'absolutely' minimal programs, we now look at 'relatively' minimal programs. Given a set $\{\varphi_e\}_{e \in A}$ of partial recursive functions, the set of *programs minimal w.r.t. A* is defined as follows:

$$M_A = \{e \in A : (\forall i \in A)[\varphi_e \simeq \varphi_i \ \Rightarrow\ m(e) \leq m(i)]\}.$$

Thus $M_\omega = M$.

Corollary VII.1.7 *If M_A is r.e. and infinite, $\{\varphi_e\}_{e \in A}$ contains partial recursive functions which admit recursive programs arbitrarily smaller in size than any program for them in A.*

Proof. This is just a special case of the Size Shrinkage Theorem, obtained by letting $B = M_A$ in its statement. $\square$

The next exercise gives a wide range of applications.

Exercise VII.1.8 *For any recursive set A such that*

$$e \in A \Rightarrow \varphi_e \text{ total,}$$

the set M_A of programs minimal w.r.t. A is r.e. (Hint: given x find effectively

$$I_x = \{y : m(y) < m(x)\}.$$

Then

$$x \in M_A \Leftrightarrow x \in A \wedge (\forall y \in I_x \cap A)(\exists z)(\varphi_x(z) \neq \varphi_y(z)).$$

Then M_A is r.e., since φ_x and φ_y are total if $x, y \in A$.)

As an example, if A is the set of primitive recursive programs and m is the identity function, by VII.1.7 and VII.1.8 there are primitive recursive functions which admit general recursive programs much smaller than the smallest primitive recursive program for them. Thus the μ-operator is not needed for the description of a primitive recursive function, but its use may allow much shorter descriptions of such a function than any possible description not using it.

In general terms, one can restate VII.1.7 by saying that *powerful tools which are not necessary for some relatively simple task may nevertheless be useful to provide better solutions*. Of course, at this stage 'better' means only 'of smaller size', since we have not yet considered the question of efficiency (i.e. dynamic measures). After VII.1.19, 'better' will also mean 'of smaller size and with a fixed efficiency loss'.

Constable [1971], Meyer and Bagchi [1972], Cook and Borodin have shown that *the Size Shrinkage Theorem also holds for subrecursive programs*, in the sense that given any two subrecursive classes of functions such that the second contains a universal function for the first, then there are functions admitting programs in the second class arbitrarily smaller than any program in the first. For example, there are elementary functions (see Section VIII.7) which admit primitive recursive programs much smaller than the smallest elementary program for them.

Exercises VII.1.9 Minimal indices and enumerations without repetitions. In the following exercises we consider the special case of the measure $m(e) = e$.

a) *For any recursive set A such that*

$$e \in A \;\Rightarrow\; \varphi_e \textit{ total,}$$

the class $\{\varphi_e\}_{e\in A}$ can be recursively enumerated without repetitions. (Liu [1960]) (Hint: see VII.1.8.)

b) *The existence of a recursive enumeration without repetitions of the r.e. sets (II.5.22) and of the partial recursive functions (II.5.23.a) can be obtained without the priority method.* (Kummer [1989], [1990]) (Hint: consider a canonical enumerations of the finite sets with an even number of elements, as well as a recursive enumeration of the class of the remaining r.e. sets. As in VII.1.6.b, the set of minimal indices of the latter enumeration is Σ^0_2, which means that the class itself is enumerable without repetitions, recursively in $\mathcal{K}$. Let $f(e,n)$ be a recursive approximation of such an enumeration, by the Limit Lemma. In the terminology of II.5.22, if x is a follower of e and $f(e,n) = f(e,n+1)$, let $S_{x,n+1} = \mathcal{W}_{e,n+1}$. If instead $f(e,n) \neq f(e,n+1)$, release x, let $S_{x,n}$ be equal to an unused finite set with an even number of elements containing $S_{x,n}$, and erase such a set from the canonical enumeration. Moreover, at each step enumerate the first unused finite set with an even number of elements, and erase such a set from the canonical enumeration.)

Exercises VII.1.10 Static complexity of arbitrary sets. Given a static measure m and a set A, we define the **static complexity of a set A** as the total and nondecreasing function

$$m_A(x) = \min\{m(e) : (\forall y \le x)(\varphi_e(y) \simeq c_A(y))\}.$$

Thus $m_A(x)$ measures the minimal complexity of partial recursive functions agreeing with c_A on the initial segment up to x (Bardzin [1968], Kanovich [1969], Loveland [1969]).

a) *A set A is recursive if and only if m_A is bounded.* (Kanovich [1969], Loveland [1969]) (Hint: if A is recursive then $c_A \simeq \varphi_e$ for some e, and m_A is bounded by $m(e)$. Conversely, if m_A is bounded, then it is constant from a certain point on because it is nondecreasing. But there are only finitely many recursive functions with a given complexity, and then there is a recursive function agreeing with c_A on arbitrarily long segments, and hence always.)

b) *An r.e. set A is wtt-complete if and only if m_A majorizes an unbounded (nondecreasing) recursive function.* (Kanovich [1969], [1970a]) (Hint: by III.8.17, A is *wtt*-complete if and only if there is a function $f \le_{wtt} A$ without fixed-points. Suppose m_A majorizes some recursive and unbounded function g. For any recursive function h,

$$\mathcal{W}_{f(x)} = \overline{A} \cap \{0, \ldots, h(x)\}$$

defines a function $f \leq_{wtt} A$, and therefore it suffices to choose h such that, for every x, $\mathcal{W}_{f(x)} \neq \mathcal{W}_x$. If $\mathcal{W}_{f(x)} = \mathcal{W}_x$ for any h, then the function

$$\varphi_{t(x)}(z) \simeq \begin{cases} 1 & \text{if } z \text{ shows up first in } A \\ 0 & \text{if } z \text{ shows up first in } \mathcal{W}_x \end{cases}$$

agrees with c_A up to $h(x)$. Therefore, for all $z \leq h(x)$,

$$g(z) \leq m_A(z) \leq m(t(x)).$$

Since g is recursive and unbounded, effectively in x we can find some number z_x such that $m(t(x)) < g(z_x)$. Thus it is enough to let $h(x) = z_x$ for all x.

Conversely, let A be *wtt*-complete and let $f \leq_{wtt} A$ be without fixed-points. Let $f \simeq \varphi_i^A$ with bound h. Given e, consider $\varphi_i^{\varphi_e}$ and effectively obtain a fixed-point z for it. Since f has no fixed-points, $f(z) \not\simeq \varphi_i^{\varphi_e}(z)$ and hence φ_e and c_A must differ below $h(z)$. Thus, given e, we can effectively find an upper bound of the elements on which φ_e and c_A agree. To define g recursive and unbounded such that m_A majorizes g, let

$$g(x) = 0 \text{ for all } x \leq x_0,$$

where x_0 is a bound to the elements on which φ_e and c_A agree, for every e such that $m(e) = 0$: if φ_e and c_A agree up to x_0, then $m(e) > 0$. We can continue by defining

$$g(x) = 1 \text{ for all } x_0 < x \leq x_1,$$

for $x_1 > x_0$ defined in a similar way for $m(e) = 1$, and so on.)

c) *An r.e. set A is T-complete if and only if m_A majorizes an unbounded (nondecreasing) function recursive in A.* (Kanovich [1970a]) (Hint: similar to part b), using III.1.5 in place of III.8.17.)

While for any nonrecursive set A the function m_A does grow to infinity, if A is r.e. and not too complicated (i.e. not *wtt*- complete), this growth is *quite slow* (in the sense that any unbounded recursive function is greater infinitely often). In particular, each hypersimple set (being not *wtt*-complete by III.8.16) has such a slow growth of static complexity.

Daley [1976] has characterized the class of the r.e. sets A such that the growth of m_A is *very slow* (in the sense that m_A is dominated by every unbounded nondecreasing recursive function).

For more on static measures, see Meyer [1972], Kinber [1977], Marandzjan [1977], [1979].

Shortening proofs by adding axioms $\star$

Speed-up phenomena were originally discovered in the context not of speed of computations, but of length of proofs. The next result provides an abstract formulation of a speed-up phenomenon for formal systems.

Theorem VII.1.11 Gödel's Speed-Up Theorem (Gödel [1936]) *Let $\mathcal{F}^* \supseteq \mathcal{F}$ be formal systems (with recursive sets of axioms and of recursive rules) such that $\mathcal{F}^* - \mathcal{F}$ is not r.e. Given a recursive factor h, there is a theorem φ of $\mathcal{F}$ and a number n such that φ admits a proof of length $\leq n$ in $\mathcal{F}^*$, but no proof of length $\leq h(n)$ in $\mathcal{F}$.*

Proof. Suppose that, for all n, every theorem of $\mathcal{F}$ with a proof of length $\leq n$ in $\mathcal{F}^*$ has a proof of length $\leq h(n)$ in $\mathcal{F}$. Then $\mathcal{F}^* - \mathcal{F}$ is r.e., as follows: generate all proofs of $\mathcal{F}^*$, and when a theorem φ with a proof of length n is found, one can decide whether φ is already a theorem of $\mathcal{F}$ by checking every proof of length $\leq h(n)$ in $\mathcal{F}$. □

We now give a general condition under which the hypotheses of the result just proved are satisfied. The condition is expressed in terms of the notion of essential undecidability defined in III.10.6.

Proposition VII.1.12 *If:*

1. *$\mathcal{F}$ is an essentially undecidable formal system*
2. *φ is not provable in $\mathcal{F}$*
3. *$\mathcal{F}^*$ is the logical closure of $\mathcal{F} \cup \{\varphi\}$*

then $\mathcal{F}^* - \mathcal{F}$ is not r.e.

Proof. Since φ is not provable in $\mathcal{F}$, the logical closure of $\mathcal{F} \cup \{\neg\varphi\}$ is a consistent extension of $\mathcal{F}$, and hence is not recursive by the essential undecidability of $\mathcal{F}$. But

$$(\neg\psi \to \varphi) \in \mathcal{F}^* - \mathcal{F} \Leftrightarrow \text{ not } (\vdash_{\mathcal{F}\cup\{\neg\varphi\}} \psi),$$

because $\neg\psi \to \varphi$ (i.e. $\psi \vee \varphi$) is always in $\mathcal{F}^*$ (since so is φ), and it is in $\mathcal{F}$ if and only if $\neg\varphi \to \psi$ is, i.e. if and only if $\vdash_{\mathcal{F}\cup\{\neg\varphi\}} \psi$ (by the Deduction Theorem).

If $\mathcal{F}^* - \mathcal{F}$ were r.e., then the closure of $\mathcal{F} \cup \{\neg\varphi\}$ would be recursive, since its complement would be r.e. □

Thus adding an unprovable sentence to an essentially undecidable formal system $\mathcal{F}$ radically shortens some proof of some theorem of $\mathcal{F}$.

Notice that this can always be done for any consistent formal system $\mathcal{F}$ extending $\mathcal{R}$ (defined in I.3.6), since $\mathcal{F}$ is essentially undecidable (by III.10.11) and it admits unprovable sentences (by II.2.17.2).

For more on the subject of Gödel's speed-up see Mostowsky [1952], Ehrenfeucht and Mycielsky [1971], Parikh [1971], [1973], and Statman [1978], [1981].

Definition of a dynamic complexity measure

The next definition imposes only minimal conditions on the notion of dynamic measure. We discuss below the advantages and disadvantages of such a generality.

Definition VII.1.13 (Blum [1967]) *Given an acceptable system $\{\varphi_e\}_{e\in\omega}$ for the partial recursive functions, a* **dynamic complexity measure** *is a family $\{\Phi_e\}_{e\in\omega}$ of partial recursive functions such that:*

1. *for every e and x, $\Phi_e(x)\downarrow$ if and only if $\varphi_e(x)\downarrow$*
2. *the predicate $\Phi_e(x) \simeq z$ is recursive (uniformly in e, x and z).*

Φ_e is variously referred to as a **resource measure**, a **complexity**, a **step-counting function**, or a **running time** for φ_e.

The important part of Axiom 1 is that if $\varphi_e(x)\downarrow$, then $\Phi_e(x)\downarrow$, i.e. only a finite resource amount is needed in a convergent computation. The converse is not so crucial, since it can be trivially ensured by defining a measure only when the computation converges (see the case of space complexity below). A weakening of Axiom 1, not requiring $\Phi_e(x)$ to diverge when $\varphi_e(x)$ does, has been considered by Ausiello [1971] and Gill and Simon [1976].

Axiom 2 captures the intuition that the restriction to a finite resource amount z forces the computation of $\varphi_e(x)$ to either converge or to have a periodic behavior, and that one can effectively differentiate these two cases by just running the computation within the given resource amount.

We use the following **convention**:

> $\Phi_e(x) = \infty$ when $\varphi_e(x)\uparrow$, where ∞ is a symbol such that $n < \infty$ and $\infty \leq \infty$.

Thus $\Phi_e(x) > z$ does not imply that $\varphi_e(x)$ converges, while $\Phi_e(x) \leq z$ does.

As a first example, one can measure the resource of $\varphi_e(x)$ by the smallest code of its computations. Formally:

$$\Phi_e(x) \simeq \mu y.\, \mathcal{T}_1(e,x,y),$$

where $\mathcal{T}_1$ is the predicate of the Normal Form Theorem II.1.2. Since every approach to recursiveness can be arithmetized and produces a version of the Normal Form Theorem, a whole family of dynamic measures is thus obtained.

More specific and popular measures, studied in detail in Section 4 and in Chapter VIII, refer to Turing machines as follows:

- *space complexity* (Trakhtenbrot [1956], Myhill [1960], Ritchie [1963], Hartmanis, Lewis and Stearns [1965])

  $$S_e(x) \simeq \begin{cases} \text{number of cells scanned in computing } \varphi_e(x) & \text{if } \varphi_e(x)\downarrow \\ \text{undefined} & \text{otherwise.} \end{cases}$$

  Note that a Turing machine can scan a finite number of cells even when the computation is undefined (by cycling), but we can effectively notice when a cycle has happened, and let $\Phi_e(x)$ diverge in this case.

- *time complexity* (Yamada [1962], Rabin [1963], Hartmanis and Stearns [1965])

  $$T_e(x) \simeq \text{ number of moves performed in computing } \varphi_e(x).$$

These are by no means the only possible measures that can be associated with Turing machine computations. Other sensible alternatives would be to count the number of *reversals* in the direction of the head moves, the maximum number of *crossings* of the boundaries between two tape cells (see VII.4.7.a for a proof using this measure), the maximum number of *returns* to a given cell before the final (alternatively, after the first) alteration of its content, the amount of *ink* used (if one assumes that a constant amount of ink is used each time a symbol is changed on the tape, this is measured by the number of changes performed). Results relative to these measures are reported in Wagner and Wechsung [1986].

As a final example, a measure can be defined by taking any measure $\{\Phi_e\}_{e\in\omega}$ and any recursive function f such that $f(x) \geq x$ for every x, and by considering $\{f \circ \Phi_e\}_{e\in\omega}$. The first axiom of VII.1.13 is trivially verified. For the second axiom, to check whether $f(\Phi_e(x)) \simeq z$ holds, we only have to check whether, for some $y \leq z$, $\Phi_e(x) \simeq y$ and $f(y) = z$.

Exercises VII.1.14 a) *The predicates $\Phi_e(x) \leq z$ and $\Phi_e(x) \geq z$ are both recursive.* Of course, only the first implies $\varphi_e(x)\downarrow$.

b) *The two axioms of VII.1.13 are independent.* (Blum [1967]) (Hint: $\Phi_e \simeq \varphi_e$ satisfies the first axiom but not the second, since not every partial recursive function has a recursive graph. $\Phi_e(x) \simeq e$ satisfies the second axiom but not the first.)

c) *There are partial recursive functions which are not step-counting functions for any measure.* (Hint:

$$\varphi(x) \simeq 0 \;\Leftrightarrow\; x \in \mathcal{K}$$

does not have a recursive graph.)

d) *Given any measure, there are total recursive functions which are not step-counting functions w.r.t. that measure.* (Blum [1967]) See VII.2.10.c and VII.3.10 for improvements. (Hint: consider

$$f(x) = \begin{cases} 0 & \text{if } \Phi_x(x) \simeq 1 \\ 1 & \text{otherwise.)} \end{cases}$$

e) *Given any two recursive functions f and g, there are measures in which f has g as its best complexity.* (Hint: given a measure $\{\Phi_e\}_{e\in\omega}$ and an index i for f, let

$$\Psi_e(x) \simeq \begin{cases} g(x) & \text{if } e = i \\ \Phi_e(x) + g(x) & \text{otherwise.} \end{cases}$$

Then $\{\Psi_e\}_{e\in\omega}$ is still a measure.)

f) *Given any recursive set A and any recursive function f such that*

$$e \in A \;\Rightarrow\; \varphi_e \textit{ and } \varphi_{f(e)} \textit{ are total,}$$

there are measures in which, for every $e \in A$, φ_e has complexity $\varphi_{f(e)}$. (Hint: given any measure $\{\Phi_e\}_{e\in\omega}$, let

$$\Psi_e \simeq \begin{cases} \varphi_{f(e)} & \text{if } e \in A \\ \Phi_e & \text{otherwise.)} \end{cases}$$

Thus one can change a given measure by arbitrarily assigning complexities to infinitely many programs for total functions.

The notion of measure introduced in VII.1.13 is very general and the two axioms, being very natural, are likely to hold for every *natural measure*. An obvious advantage of such a generality is that results that hold for every measure (and, as we will see in Sections 2 and 3, a great deal of theory can indeed be developed for general measures) also apply to every natural measure.

Because of its generality, however, Definition VII.1.13 also subsumes *unnatural measures*. Thus purely existential results (asserting the existence of measures with certain properties) are not likely to be very useful without further specifications. Moreover, because of results such as VII.1.14.e, the abstract theory of complexity is unlikely to have applications in the study of specific functions (since every function is trivial to compute in some measures).

Some efforts have been devoted to the search of axioms which, added to those of VII.1.13, would exclude unnatural measures. See, for example, Arbib and Blum [1965], Burkhard and Kroon [1971], Hartmanis [1973], Alton [1980], and Wagner and Wechsung [1986] (see also Havel [1971] for a different alternative). Some proposals that might lead to new axioms are:

- The complexities of programs that differ only notationally should be the same.
- The complexity of a program obtained by adding a finite table (that gives a finite number of values for free) to another program should not be much larger than the complexity of the original program (*finitely invariant measures*, see Lewis [1971a] and Borodin [1972]). Some slight increase in complexity should be expected everywhere, due to the need for checking, for a given argument, whether the table is applicable or not.

- The complexity of a program built up from subprograms should be naturally related to the complexities of the subprograms.

  For example, one might request the existence of a recursive function f such that, for all i and e,

  $$\varphi_{f(i,e)}(x) \simeq \begin{cases} \varphi_i(x) & \text{if } \Phi_i(x) \leq \Phi_e(x) \\ \varphi_e(x) & \text{otherwise} \end{cases}$$

  and

  $$\Phi_{f(i,e)}(x) = \min\{\Phi_i(x), \Phi_e(x)\}$$

  (*parallel computation property*, see Landweber and Robertson [1972]).

  In a similar spirit, one might request the existence of recursive functions g and h such that, for all i and e,

  $$\varphi_{g(i,e)}(x) \simeq \varphi_i(\varphi_e(x))$$

  and

  $$\Phi_{g(i,e)}(x) \leq h(\Phi_e(x), \Phi_i(\varphi_e(x)))$$

  (*compositionality*, see Lischke [1975], [1976], [1977]).

- The resource amount needed to determine the complexity of $\varphi_e(x)$ should be related to the resource amount needed to compute $\varphi_e(x)$, e.g. there should be a recursive function f and (for example) a constant c such that

  $$\varphi_{f(e)} \simeq \Phi_e$$

  and

  $$(\forall_\infty x)[\Phi_{f(e)}(x) \leq c \cdot \Phi_e(x)].$$

Finally, Definition VII.1.13 is not readily applicable to *subrecursive programs*, as was VII.1.2. For example, one might think of considering a system of indices $\{\psi_e\}_{e\in\omega}$ for the primitive recursive functions and, as in VII.1.13, call $\{\Psi_e\}_{e\in\omega}$ a measure for it if:

- for each e, Ψ_e is total
- the predicate $\Psi_e(x) = z$ is primitive recursive (uniformly in e, x and z).

But then this would allow absolutely trivial measures, such as

$$\Psi_e(x) = 0 \text{ for all } e \text{ and } x.$$

More axioms are thus definitely needed in this context, and the topic is discussed in Constable and Borodin [1972], Alton [1980], Kozen [1980]. Part of the problem comes from the fact that we do not yet have a satisfactory notion of an acceptable system of indices for subclasses of recursive functions.

First properties of dynamic complexity measures

The next result is an analogue of VII.1.3, and presents for the first time two common features of this chapter:

- The statement refers not to 'all', but only to 'almost all' numbers. This is natural when dealing with complexities, since one can always add to a program a finite table that gives a finite number of values for free (thus, when dealing with *upper bounds* of some complexity of a given function, it is enough to provide them for almost all arguments, and when dealing with *lower bounds* of all complexities of a given function, it would make no sense to look for nontrivial ones that would work on all arguments).
- The proof uses the **method of maximization**, which first provides a bound working for all x but depending (uniformly) on e, and then takes the maximum over $e \leq x$, thus obtaining a bound independent of e but working, for each e, only on almost all x.

Proposition VII.1.15 Recursive Relatedness of Dynamic Measures (Blum [1967]) *Given any two dynamic measures $\{\Phi_e\}_{e\in\omega}$ and $\{\Psi_e\}_{e\in\omega}$ (for the same acceptable system of indices) there is a recursive function g such that, for every e and almost every x,*

$$\Phi_e(x) \leq g(x, \Psi_e(x)) \qquad \textit{and} \qquad \Psi_e(x) \leq g(x, \Phi_e(x)).$$

Proof. Let

$$h(e,x,z) = \begin{cases} \max\{\Phi_e(x), \Psi_e(x)\} & \text{if } \Phi_e(x) \simeq z \text{ or } \Psi_e(x) \simeq z \\ 0 & \text{otherwise.} \end{cases}$$

The function h is recursive, as follows. $\Phi_e(x) \simeq z$ and $\Psi_e(x) \simeq z$ are recursive by Definition VII.1.13. If one of them holds, then $\varphi_e(x)\downarrow$, i.e. both $\Phi_e(x)\downarrow$ and $\Psi_e(x)\downarrow$.

If $\varphi_e(x)\downarrow$, we then have

$$h(e,x,\Psi_e(x)) = \max\{\Phi_e(x), \Psi_e(x)\} \geq \Phi_e(x).$$

Similarly with Φ_e and Ψ_e interchanged. This provides a relatedness function that depends on e. The dependency can easily be avoided by letting

$$g(x,z) = \max\{h(e,x,z) : e \leq x\}.$$

If $x \geq e$ and $\varphi_e(x)\downarrow$, we then have

$$g(x, \Psi_e(x)) \geq h(e,x,\Psi_e(x)) \geq \Phi_e(x).$$

Similarly with Φ_e and Ψ_e interchanged. This provides a relatedness function independent of e, but working only for $x \geq e$ (hence only for almost all x). □

Exercises VII.1.16 a) *In VII.1.15 the two measures do not need to refer to the same acceptable system of indices.* (Hint: use the fact that two acceptable systems of indices are recursively isomorphic, see II.5.7.)

b) *The recursive function g providing relatedness of two measures cannot, in general, be unary.* (Hint: let $\Phi_e = T_e$ and $\Psi_e = S_e$, and suppose there is a recursive unary function g such that $\Phi_e(x) \leq g(\Psi_e(x))$ for all e and almost every x. Then any algorithm that runs in constant space would also run in constant time, contradicting VIII.1.7.)

c) *If, for every x, $\Phi_e(x) \geq x$ and $\Psi_e(x) \geq x$, then there is a unary recursive function providing relatedness.* (Hint: consider

$$\max\{h(e,x,z) : e \leq x \leq z\}$$

in the proof above.)

d) *There is no fixed recursive function g which gives relatedness of every pair of measures, even for a fixed acceptable system.* (Hint: suppose $\Phi_e(x) \leq g(x, \Psi_e(x))$ holds for any pair of measures. We may suppose $g(x,y) \geq y$, by possibly considering $g(x,y)+y$. But then, given a measure Ψ_e, the following is still a measure:

$$\Phi_e(x) \simeq g(x, \Psi_e(x)) + 1,$$

which is a contradiction.)

It follows from the exercises above that not only are any two measures associated with the same method of computation (i.e. with the same acceptable system of indices) related, but also any two measures associated with any two methods of computation are related.

Of course, both the factor g and the number of possible exceptions can be arbitrarily large, and thus *the practical interest of relatedness is quite limited.*

One possible use of relatedness of measures is to show that results are measure-independent. This is particularly useful when a certain result admits simple proofs for particular measures, since measure-independence then implies that the result holds for all measures. An example is given in the first proof of Blum's Speed-Up Theorem VII.2.16.

Exercises VII.1.17 Measure-independent properties. Let $\mathcal{L}$ be a first-order language with equality, quantifiers $\forall_\infty$ and $\exists_\infty$, no individual constants, and two predicate constants p and P with the following interpretation:

$$p(e,x,z) \Leftrightarrow \varphi_e(x) \simeq z \qquad \text{and} \qquad P(e,x,z) \Leftrightarrow \Phi_e(x) \simeq z.$$

Validity in $\mathcal{L}$ is intended as truth in every model, i.e. for every acceptable system $\{\varphi_e\}_{e\in\omega}$ and measure $\{\Phi_e\}_{e\in\omega}$. A sentence of $\mathcal{L}$ is **recursion-theoretic** if it does not contain P, and **measure-independent** if its truth does not depend on the measure used for the interpretation of P.

a) *The truth of recursion-theoretic sentences does not depend on the acceptable system used in the interpretation of p.* (Blum and Gill [1973]) (Hint: use the fact that two acceptable systems of indices are recursively isomorphic, see II.5.7.)

b) *A sentence is measure-independent if and only if it is (logically equivalent to) a recursion-theoretic sentence.* (Bennison [1980]) (Hint: since any measure is an r.e. sequence of functions $\{\varphi_{h(e)}\}_{e\in\omega}$ such that $\varphi_{h(e)}$ has domain $\mathcal{W}_e$ and the graph of $\varphi_{h(e)}$ is recursive, uniformly in e, substitute any occurrence of $P(e,x,z)$ in a given sentence by an occurrence of $\varphi_{\varphi_i(e)}(x) \simeq z$, and add a conjunct which says that i and j exist such that $\{\varphi_{\varphi_i(e)}\}_{e\in\omega}$ is a measure, i.e.

$$\varphi_{\varphi_i(e)}(x)\downarrow \Leftrightarrow \varphi_e(x)\downarrow \qquad \text{and} \qquad \varphi_j(e,x,z) \simeq x \;\Leftrightarrow\; \varphi_{\varphi_i(e)}(x) \simeq z.$$

Note that we do not simply use characteristic functions, because constants are not available. If i and j are as above, then $\{\varphi_{\varphi_i(e)}\}_{e\in\omega}$ is a measure. And the new sentence is equivalent to the original one, by measure-independence.)

c) *Measure-independence is an undecidable property.* (Bennison [1980]) (Hint: let α be a measure-independent sentence, and β be a measure-dependent sentence. If γ is a recursion-theoretic sentence, let

$$\gamma^* \;\Leftrightarrow\; (\gamma \wedge \alpha) \vee (\neg\gamma \wedge \beta).$$

If γ is true, γ^* is equivalent to α and hence is measure-independent. If γ is false, then γ^* is measure-dependent. Thus, if measure-independence were decidable, so would be truth for recursion-theoretical sentences, which is not.)

The statement of the Fixed-Point Theorem II.2.10 hides a number of features implicit in its proof. One was made explicit in II.3.16, where the fact that the computation of $\varphi_e(x)$ takes longer than the computation of $\varphi_{f(e)}(x)$ was exploited to show that the fixed-point provided by the proof of II.2.10 actually is the *least* fixed-point. The next result provides a complement to that analysis, and shows that the complexities of $\varphi_e(x)$ and $\varphi_{f(e)}(x)$ are recursively related, in a uniform way.

Theorem VII.1.18 Complexity-Theoretic Fixed-Point Theorem (Blum [1971]) *Given a measure $\{\Phi_e\}_{e\in\omega}$, there is a recursive function g (depending only on the measure) such that, for every recursive function f, there is e such that*

1. $\varphi_e \simeq \varphi_{f(e)}$
2. $\Phi_e(x) \leq g(x, \Phi_{f(e)}(x))$, *for almost every x.*

Proof. Since we want g to depend only on the measure and not on f itself, we use the following uniform version of the Fixed-Point Theorem (see p. I.155): there exists a total recursive function h such that

$$\varphi_a \text{ total} \;\Rightarrow\; \varphi_{h(a)} \simeq \varphi_{\varphi_a(h(a))}$$

(i.e. h gives a fixed-point of φ_a uniformly in a, when φ_a is total). Let

$$t(a,x,z) = \begin{cases} \Phi_{h(a)}(x) & \text{if } \Phi_a(h(a)) \leq x \ \wedge \ \Phi_{\varphi_a(h(a))}(x) \simeq z \\ 0 & \text{otherwise} \end{cases}$$

and

$$g(x,z) = \max\{t(a,x,z) : a \leq x\}.$$

Notice that t is well-defined. Indeed, if $\Phi_a(h(a)) \leq x$, then we know that $\varphi_a(h(a))\downarrow$, and we can check whether $\Phi_{\varphi_a(h(a))}(x) \simeq z$. Moreover, by definition of h, if $\varphi_{\varphi_a(h(a))}(x)\downarrow$, then $\varphi_{h(a)}(x)\downarrow$.

Given f recursive, let $f \simeq \varphi_a$ and $e = h(a)$. Since f is total, $\varphi_e \simeq \varphi_{f(e)}$ by the choice of h. And if $x \geq \Phi_a(e)$ and $x \geq a$, then

$$g(x, \Phi_{f(e)}(x)) \geq t(a, x, \Phi_{f(e)}(x)) = \Phi_e(x). \quad \Box$$

Exercise VII.1.19 Size-speed trade-off. (Blum [1967a]) *Given static and dynamic measures m and $\{\Phi_e\}_{e\in\omega}$, the Size Shrinkage Theorem VII.1.4 can be improved by the clause that the loss in efficiency is limited by a recursive factor (depending only on the measures).* Precisely, there is a recursive function g such that, for every recursive factor h and any infinite r.e. set of programs B, we can find a program $e \in B$ and another program i computing the same function, such that

$$h(m(i)) \leq m(e) \qquad \text{and} \qquad \Phi_i(x) \leq g(x, \Phi_e(x)), \text{ for almost every } x.$$

(Hint: use VII.1.18 in the proof of VII.1.4, in place of the usual Fixed-Point Theorem.)

For more results on size-speed trade-off, see Constable [1971], and Constable and Borodin [1972].

VII.2 Speed of Computations

Since our main interest is in sets of natural numbers, we pay special attention to characteristic (0,1-valued) functions. In particular, in this section we study the complexity of total recursive functions, and leave the study of the complexity of partial recursive functions to Section IX.4.

Upper and lower bounds of the complexities of a function

Definition VII.2.1 *Given a recursive function f, we say that a function t is:*

1. *an* **upper bound** *of (some of) the complexities of f if t dominates the complexity of at least one program for f, i.e.*

$$(\exists e)[f \simeq \varphi_e \ \wedge \ (\forall_\infty x)(\Phi_e(x) \leq t(x))]$$

2. *a* **lower bound** *of (all) the complexities of f if t is dominated by the complexities of every program for f, i.e.*

$$(\forall e)[f \simeq \varphi_e \Rightarrow (\forall_\infty x)(t(x) \leq \Phi_e(x))].$$

Thus t is an upper bound of the complexities of f if there is at least one method of computing f that does not take more than t, and is a lower bound if every method of computing f takes at least t. In both cases, as already discussed in Section 1, we allow for finitely many exceptions (the intuition being that finitely many values of a function can always be incorporated in a finite table, and thus provided at practically no cost).

The smaller the difference between upper and lower bounds, the better will be the characterization of the 'best complexity' of a given function.

Exercise VII.2.2 Slow-Down Theorem. *Given $\{\Phi_e\}_{e\in\omega}$, for any pair of recursive functions f and t there is one index e of f such that, for every x, $t(x) \leq \Phi_e(x)$.* (Blum [1967]) (Hint: let

$$\varphi_{h(e)}(x) \simeq \begin{cases} \varphi_e(x)+1 & \text{if } \Phi_e(x) < t(x) \\ f(x) & \text{otherwise,} \end{cases}$$

and use the Fixed-Point Theorem. Or prove the result for a measure such as time, for which it is trivial, and then use VII.1.15.)

While the exercise shows that there can be no notion of an upper bound of *all* the complexities of a recursive function, the next result shows that the notion of a lower bound is not trivial. The proof contains ideas that are thoroughly exploited throughout this section.

Theorem VII.2.3 Rabin's Theorem (Rabin [1960a], Tseitin) *Every recursive function t is the lower bound of the complexities of some 0,1-valued recursive function f. Precisely, given t, there is a 0,1-valued recursive function f such that*

$$(\forall e)[f \simeq \varphi_e \Rightarrow (\forall_\infty x)(t(x) \leq \Phi_e(x))].$$

Proof. We break down the global condition on f into an infinite sequence of local requirements:

$$R_e \ : \ f \simeq \varphi_e \Rightarrow (\forall_\infty x)(t(x) \leq \Phi_e(x)).$$

These are not easy to satisfy as they stand, since their premises talk about the very function we have to build. For the purpose of a construction of f, it is better to deal with their (logically equivalent) contrapositives:

$$R_e \ : \ (\exists_\infty x)(\Phi_e(x) < t(x)) \Rightarrow f \not\simeq \varphi_e.$$

The strategy for a single requirement R_e is simple: wait for a stage x when $\Phi_e(x)$ drops below $t(x)$ (in particular, $\varphi_e(x)\downarrow$), and then let $f(x) \not\simeq \varphi_e(x)$. Putting together the strategies would be immediate if we could recursively enumerate the requirements for which the hypothesis holds (the only ones for which we need to take action, the others being trivially satisfied): then we could satisfy the requirements in order (by letting f be equal to 0 at waiting stages). But the hypothesis on R_e is infinitary and not recursively testable, and so we cannot get rid of the requirements for which it does not hold. Then, instead of satisfying only the requirements we care about, we satisfy more requirements (those for which Φ_e drops below t sufficiently often), and we do so as soon as the opportunity appears. If at a given stage we can satisfy more than one requirement, then we go back to the previous idea of satisfying the one that comes first in the natural ordering, thus ensuring that eventually every requirement is satisfied.

The construction is as follows. At stage x we define $f(x)$. Consider e such that:

- $e \leq x$
- R_e has not yet been satisfied, i.e. e has not been cancelled[1]
- R_e can be satisfied at this stage, i.e. $\Phi_e(x) < t(x)$.

If there is such an e, choose the least one e_x, define

$$f(x) = 1 - \varphi_{e_x}(x)$$

and cancel e_x (since R_{e_x} has been satisfied). If there is no such e, let

$$f(x) = 0.$$

Notice that f is recursive: at stage x we only look for $e \leq x$, the condition $\Phi_e(x) < t(x)$ is recursive, and if it holds, then $\varphi_e(x)\downarrow$ (and hence $\varphi_e(x)$ can be recursively computed). Moreover, by definition, f is 0,1-valued and total.

It remains to show that each R_e is satisfied. Notice that each requirement is satisfied at most once, since when it is satisfied then its index is cancelled and no longer considered. Thus we can choose $x_e \geq e$ such that after stage x_e, no requirement with index smaller than e is satisfied. We may suppose that there are infinitely many x such that $\Phi_e(x) < t(x)$, otherwise R_e is trivially satisfied. Then there is $x \geq x_e$ (in particular, $x \geq e$) such that $\Phi_e(x) < t(x)$. If

[1]Instead of cancelling indices one could directly check the condition

$$(\forall y < x)[\Phi_e(y) < t(y) \ \Rightarrow\ f(y) \simeq \varphi_e(y)],$$

and consider R_e only if it holds.

R_e has not yet been satisfied (in the opposite case there is nothing to prove), it is considered at stage x and it must be $e = e_x$ (otherwise a requirement with a smaller index would be considered and satisfied at stage x, contradicting the choice of x_e). Then R_e is satisfied at stage x. $\Box$

Notice that the restriction to 0,1-valued functions makes the result not trivial. For example, in the case of time complexity, the result without the restriction could be proved by noting that the function $t(x)+x$ already requires $t(x)$ moves simply to write down the output (in unary notation). In general, for any measure $\{\Phi_e\}_{e\in\omega}$, the function

$$f(x) = \max\{\varphi_e(x) + 1 : \Phi_e(x) < t(x) \ \wedge \ e \leq x\}$$

would suffice. The restriction then ensures that the function f requires a long time to compute because it is difficult to *find* its values, and not simply because such values are difficult to *write*.

The proof of the result above uses diagonalization (locally, to satisfy a single requirement), and a **priority argument with no injury**, which consists simply of satisfying once and for all the first requirement that has not yet been satisfied and that can be satisfied (globally, to take care of all the requirements). Priority arguments of this kind are also called **cancellation arguments**. They have been already used in Chapter III (see, for example, the proofs of III.2.11 and III.3.12), and are the main tool of this chapter. The priority method is discussed in full generality in Chapter X.

Exercise VII.2.4 *Rabin's Theorem does not hold constructively, in the sense that if t is sufficiently large, then for no 0,1-valued recursive function f such that, for all e,*

$$f \simeq \varphi_e \ \Rightarrow \ (\forall_\infty x)(t(x) \leq \Phi_e(x)),$$

there is a partial recursive function ψ such that, for all e,

$$f \simeq \varphi_e \ \Rightarrow \ \psi(e)\!\downarrow \wedge\, (\forall x \geq \psi(e))(t(x) \leq \Phi_e(x)).$$

(Fulk [1990a]) (Hint: given a recursive, one-one function g with range $\mathcal{K}$, let

$$\varphi_{t_0(e,x)}(s) \simeq \begin{cases} 0 & \text{if } g(s) = x \\ \varphi_e(s) & \text{otherwise} \end{cases} \quad \text{and} \quad \varphi_{t_1(e,x)}(s) \simeq \begin{cases} 1 & \text{if } g(s) = x \\ \varphi_e(s) & \text{otherwise.} \end{cases}$$

Suppose

$$t(z) > \max_{e,x,s\leq z}\{\Phi_{t_0(e,x)}(s), \Phi_{t_1(e,x)}(s) : g(s) = x\}$$

and let f be a 0,1-valued recursive function such that

$$f \simeq \varphi_e \ \Rightarrow \ (\forall_\infty x)(t(x) \leq \Phi_e(x)).$$

If $x \notin \mathcal{K}$, then $\varphi_{t_0(e,x)} \simeq \varphi_{t_1(e,x)} \simeq \varphi_e$. If $x \in \mathcal{K}$, then there is a unique s such that $g(s) = x$, so that only one of $\varphi_{t_0(e,x)}$ and $\varphi_{t_1(e,x)}$ is φ_e.

To decide whether $x \in \mathcal{K}$, see what happens first: if $x \in \mathcal{K}$, then we know the answer; if both $\psi(t_0(e,x))$ and $\psi(t_1(e,x))$ converge, then $x \in \mathcal{K}$ if and only if x is generated by g at a stage smaller than one of them.)

Exercises VII.2.5 (Smith [1988]) a) *Every recursive function t is the lower bound of the complexities of some 0,1-valued recursive function f and of all functions agreeing with f almost everywhere (i.e. its finite variants).* (Hint: diagonalize infinitely often against each e such that the hypothesis of R_e is satisfied. Use a counter C_e, initialized with value e at stage e, to keep track of how many times one has diagonalized against e. At any stage, diagonalize against the e with the lowest counter and, if there are many with the same counter, against the one with the lowest index among them, and increase the counter by one.)

b) *If restricted to 0,1-valued functions, the result fails for functions agreeing with f infinitely often, even if one requires the complexities to be above t only infinitely often.* (Hint: the resulting f would agree infinitely often with one of the constant functions with value 0 or 1, and there are measures in which these two functions have constant complexity.)

c) *The result holds in general for functions agreeing with f infinitely often, if one requires the complexities to be above t only infinitely often.* (Hint: let $f(x)$ be the smallest element not in the set $\{\varphi_e(x) : \Phi_e(x) < t(x) \wedge e \le x\}$.)

Exercises VII.2.6 Complex recursive functions. Given a recursive function t, we say that a recursive function f is **more complex** than t if each program for f takes more than t infinitely often, i.e.

$$(\forall e)[f \simeq \varphi_e \;\Rightarrow\; (\exists_\infty x)(t(x) < \Phi_e(x))].$$

Similarly for recursive sets (using characteristic functions).

a) *The class of functions more complex than t is not recursively enumerable.* (Gill and Blum [1974]) (Hint: given any r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$ of total recursive functions, build f recursive such that

$$R_e \;:\; (\forall_\infty x)(\Phi_e(x) \le t(x)) \;\Rightarrow\; f \not\simeq \varphi_e \qquad \text{and} \qquad P_e \;:\; f \not\simeq \varphi_{h(e)}.$$

Alternatively, the class of functions which are not more complex than t is obviously r.e. Thus, if also the class of functions which are more complex than t were r.e., so would be the class of all total recursive functions, contradicting II.2.1.)

b) *For every recursive function t and every infinite recursive set A, there is an infinite recursive subset B of A more complex than t.* (Flajolet and Steyaert [1974]) (Hint: modify VII.2.3 by working only on the elements of A. Note that every function φ_e which is not more complex than t satisfies $(\forall_\infty x)(\Phi_e(x) \le t(x))$; thus, if φ_e is the characteristic function of an infinite subset of A,

$$(\exists_\infty x)(\Phi_e(x) \le t(x) \;\wedge\; x \in A),$$

and we can diagonalize against it by using an element of A.)

c) *There is an infinite recursive set all of whose infinite recursive subsets are more complex than* t. This is an analogue of immunity (II.6.4). (Constable [1973], Flajolet and Steyaert [1974]) (Hint: similar to part b), by building A infinite and diagonalizing against any recursive function φ_e such that

$$(\exists_\infty x)(\Phi_e(x) \leq t(x) \ \wedge \ \varphi_e(x) \simeq 1).)$$

d) *If* $\{\Phi_e\}_{e\in\omega}$ *has the parallel computation property* (see p. 22) *then, for any function* f *which is more complex than* t*, there is an infinite set of arguments (called a* **complexity core***) on which* f *is more complex than* t *almost everywhere*. (Lynch [1975]) (Hint: build $A = \{a_0 < a_1 < \cdots\}$ by stages. Let $a_0 = 0$. Given a_n, do the following. Let $z = a_n + 1$ and cancel $e \leq n$ if it is still uncancelled, $\Phi_e(z) \leq t(z)$ and $\varphi_e(z) \not\simeq f(z)$. For all the uncancelled $e \leq n$, see whether $\Phi_e(z) > t(z)$: if so, let $a_{n+1} = z$; otherwise go back to the beginning and change z into $z + 1$.

The procedure must stop, otherwise for every sufficiently large z there is an index $e \leq n$ which is never cancelled and such that $\Phi_e(z) \leq t(z)$. By construction, if z is sufficiently large and $e \leq n$ is never cancelled, then

$$\Phi_e(z) \leq t(z) \ \Rightarrow \ \varphi_e(z) \simeq f(z).$$

If we put together the programs for the e's that are never cancelled, by the parallel computation property we obtain an index i for f such that $(\forall_\infty z)(\Phi_i(z) \leq t(z))$, contradicting the hypothesis on f.)

For more on complexity cores, see Lynch [1975a], Even, Selman and Yacobi [1985], Orponen [1986], Orponen and Schöning [1986], Schöning [1986], Book and Du [1987], and Book, Du and Russo [1988].

Exercises VII.2.7 Computation by table look-up. Let $\{\sigma_n\}_{n\in\omega}$ be an enumeration without repetitions of all strings (see V.2.1), i.e. of the finite 0,1-valued functions. If h is a recursive function such that $\varphi_{h(n)} \simeq \sigma_n$, we say that h computes σ_n by **table look-up**.

a) *Every sufficiently complex 0,1-valued function is complicated, in the following sense: for almost every finite set of arguments, the computation by table look-up is more efficient than any other method that computes the whole function*. Precisely, there is a recursive function g such that, whenever f is 0,1-valued and

$$(\forall e)[f \simeq \varphi_e \ \Rightarrow \ (\forall_\infty x)(g(x) \leq \Phi_e(x))],$$

then for every e and almost every n

$$\sigma_n \subseteq f \simeq \varphi_e \ \Rightarrow \ \sum\{\Phi_{h(n)}(x) : \sigma_n(x)\downarrow\} < \sum\{\Phi_e(x) : \sigma_n(x)\downarrow\}.$$

(Young [1971], Meyer) (Hint: build g recursive such that, for every x,

$$\sum\{\Phi_{h(n)}(x) : \sigma_n(x)\downarrow\} < \sum\{g(x) : \sigma_n(x)\downarrow\}.$$

It is enough to let

$$g(x) = \max\{\sum\{\Phi_{h(n)}(z) : \sigma_n(z)\downarrow\} \ : \ dom(\sigma_n) \subseteq \{0,\dots,x\} \wedge \sigma_n(x)\downarrow\}.$$

Notice that g is well-defined, because there are only finitely many strings with domain contained in $\{0,\dots,x\}$.)

b) *The result fails for static measures.* Precisely, if f has an infinite domain, then for every e and almost every n

$$\sigma_n \subseteq f \simeq \varphi_e \ \Rightarrow \ m(e) < m(h(n)).$$

(Hint: there are only finitely many i such that $m(i) \leq m(e)$.)

The Compression Theorem $\star$

Before attacking the central problem of which functions *have* best complexity, we dispose of a complementary one: which functions *are* best complexities. A general answer comes from a mild extension of Rabin's Theorem VII.2.3.

Theorem VII.2.8 Compression Theorem (Blum [1967]) *Every Φ_i is (uniformly in i) the best complexity of some 0,1-valued partial recursive function, within a recursive factor (dependent only on the measure).*

Precisely, there are two recursive functions f and g such that, for all e:

1. *$\varphi_{f(i)}$ has the same domain as Φ_i*
2. *$(\forall e)[\varphi_{f(i)} \simeq \varphi_e \ \Rightarrow \ (\forall_\infty x)(\Phi_i(x) \leq \Phi_e(x))]$*
3. *$(\forall_\infty x)[\Phi_{f(i)}(x) \leq g(x, \Phi_i(x))]$,*

i.e. the best complexity of $\varphi_{f(i)}$ is compressed between Φ_i and $g \circ \Phi_i$.

Proof. Both f and g are obtained at the end, the former by noticing that the proof is uniform in i, and the latter by maximization. Thus we just write Φ for Φ_i, and φ for $\varphi_{f(i)}$. The main condition to ensure is 2, and we proceed as in VII.2.3. The requirements on φ are

$$R_e \ : \ (\exists_\infty x)(\Phi_e(x) < \Phi(x)) \ \Rightarrow \ \varphi \not\simeq \varphi_e.$$

Notice that the bound Φ is not necessarily total (unlike t in VII.2.3), and thus we first have to see whether particular values converge.

The construction is as follows. At stage $s+1$ with $s = \langle x, z\rangle$, we attempt a definition of $\varphi(x)$. First see if $\Phi(x) \simeq z$. If not, do nothing. Otherwise, consider e such that:

- $e \leq x$

- e has not yet been cancelled
- $\Phi_e(x) < \Phi(x)$.

If there is such an e, choose the least one e_x, define

$$\varphi(x) = 1 - \varphi_{e_x}(x)$$

and cancel e_x. If there is no such e, let

$$\varphi(x) = 0.$$

The function φ is partial recursive, as follows: if $\Phi(x) \simeq z$, then $\Phi(x)\downarrow$, and so the condition $\Phi_e(x) < \Phi(x)$ is recursive. Moreover, by definition, φ is 0,1-valued and, since $\varphi(x)$ is defined if and only if $\Phi(x) \simeq z$ for some (unique) z, φ is well-defined and has the same domain as Φ. As in VII.2.3, the requirements R_e are satisfied.

Since the construction is uniform in $\Phi = \Phi_i$, there is a recursive function f such that $\varphi \simeq \varphi_{f(i)}$, and thus 1 and 2 are satisfied. It only remains to define g. Let

$$h(i,x,z) = \begin{cases} \Phi_{f(i)}(x) & \text{if } \Phi_i(x) \simeq z \\ 0 & \text{otherwise} \end{cases}$$

and

$$g(x,z) = \max\{h(i,x,z) : i \leq x\}.$$

For $x \geq i$, if $\Phi_i(x)\downarrow$ then

$$g(x, \Phi_i(x)) \geq h(i,x,\Phi_i(x)) = \Phi_{f(i)}(x). \quad \Box$$

We can think of the factor g as partitioning the partial recursive functions into equivalence classes, two functions being equivalent if they differ by no more than g. Then Φ_i can be thought of as best complexity (modulo g) for $\varphi_{f(i)}$, in the sense that Φ_i and $\varphi_{f(i)}$ have the same domain, Φ_i is in the equivalence class of some complexity of $\varphi_{f(i)}$ (though Φ_i need not be itself a complexity for $\varphi_{f(i)}$), and no program for $\varphi_{f(i)}$ has complexity below Φ_i (infinitely often).

The Compression Theorem holds in a more general setting, captured by the following useful recursion-theoretical notion.

Definition VII.2.9 (Blum [1967]) *An r.e. set* $\{\varphi_{h(e)}\}_{e\in\omega}$ *of partial recursive functions is a* **measured set** *if the predicate*

$$R(e,x,z) \Leftrightarrow \varphi_{h(e)}(x) \simeq z$$

is recursive, i.e. if the functions in the given set have recursive graphs, uniformly in e.

Typical examples of measured sets are the set $\{\Phi_e\}_{e\in\omega}$ of step-counting functions of any measure, and any r.e. set of total recursive functions (e.g. the set of primitive recursive functions).

On the other hand, the set of all total recursive functions is not measured (since, by II.2.1, it is not r.e.), and neither is the set of all partial functions (since, by II.2.7, the predicate $\simeq$ is not recursive).

Exercises VII.2.10 a) *If $\{\varphi_{h(e)}\}_{e\in\omega}$ is measured, then there is g recursive such that every $\varphi_{h(e)}$ is (uniformly in e) the best complexity of some 0,1-valued partial recursive function within the factor g.* (Blum [1967]) (Hint: as in VII.2.8.)

b) *If f, g and h are recursive functions such that $\varphi_{f(e)}$ has the same domain as $\varphi_{h(e)}$ and*

$$(\forall e)(\forall x)[\Phi_{f(e)}(x) \leq g(x, \varphi_{h(e)}(x))]$$

then $\{\varphi_{h(e)}\}_{e\in\omega}$ is a measured set. (Hint: to see whether $\varphi_{h(e)}(x) \simeq z$, first see whether $\Phi_{f(e)}(x) \leq g(x,z)$. If so, then we know that $\varphi_{f(e)}(x)$, and hence $\varphi_{h(e)}(x)$, converge: compute the latter value, and see if it is equal to z.)

c) *Measured sets are small, in the sense that it is possible to effectively build a recursive function not in the set.* This is a kind of effective meagerness, defined precisely in Calude [1982]. (Hint: let $\{\varphi_{h(e)}\}_{e\in\omega}$ be measured. Define $f \not\simeq \varphi_{h(e)}$ for each e, simply by seeing whether $\varphi_{h(x)}(x) \simeq 0$, and letting $f(x) = 1$ if so, and $f(x) = 0$ otherwise.)

Functions with best complexity

We turn now to a central problem of Complexity Theory, namely which functions *have* best complexity.

- We prove here that an appropriate notion of best complexity makes sense for measured sets of partial recursive functions, including the special case of any *single total recursive function* (or, more generally, any function with a recursive graph).

- In the next subsection, we show that no such notion exists for the class of *all total recursive functions*.

- In Section IX.4, we finally prove that even *single partial recursive functions* may not admit a notion of best complexity.

We start with a couple of trivial but useful observations.

Proposition VII.2.11 *There is a recursive function g such that, for every e and almost every x,*

$$\varphi_e(x)\downarrow \;\Rightarrow\; \varphi_e(x) \leq g(x, \Phi_e(x)).$$

Proof. By maximization. Let

$$h(e,x,z) = \begin{cases} \varphi_e(x) & \text{if } \Phi_e(x) \simeq z \\ 0 & \text{otherwise} \end{cases}$$

and

$$g(x,z) = \max\{h(e,x,z) : e \leq x\}.$$

If $x \geq e$ and $\varphi_e(x)\downarrow$, then

$$g(x, \Phi_e(x)) \geq h(e, x, \Phi_e(x)) = \varphi_e(x). \quad \Box$$

Thus *the values of a function cannot be much bigger than its best complexity* (when the latter exists). On the other hand, we now show that *even the best complexity of a function can be much bigger than its values.*

Proposition VII.2.12 *There is no recursive function g such that, for every e and almost every x,*

$$\varphi_e(x)\downarrow \;\Rightarrow\; \Phi_e(x) \leq g(x, \varphi_e(x)).$$

Proof. Suppose such a g exists, and let φ_e be a total 0,1-valued recursive function. Then, for almost every x,

$$\Phi_e(x) \leq \max\{g(x,0), g(x,1)\}.$$

Thus there would be a fixed recursive bound of every complexity of every 0,1-valued recursive function, contradicting VII.2.3. $\Box$

The last result suggests the following complexity-theoretical notion.

Definition VII.2.13 *If φ is partial recursive and g is total recursive, then φ is **g-honest** if*

$$(\exists e)[\varphi \simeq \varphi_e \;\wedge\; (\forall_\infty x)(\Phi_e(x) \leq g(x, \varphi_e(x))].$$

Recall that, by convention, the predicate $\leq$ holds if both sides are undefined. Thus, either $\varphi_e(x)\uparrow$, or the inequality holds with both sides defined.

Honest functions can be complex because their values are big, but not because of an intrinsic difficulty of computation (unlike the functions given by VII.2.3).

In any measure there are very honest functions. Indeed, if f is a recursive function such that $\varphi_{f(e)} \simeq \Phi_e$, then any fixed-point e of f satisfies $\varphi_e \simeq \Phi_e$, and thus the values of φ_e are exactly the amount of measure needed to compute them.

The connections between the recursion-theoretic notion of measuredness and the complexity-theoretic notion of honesty are explored in the next exercises.

Exercises VII.2.14 Honest functions. a) *Given a measured set of functions* $\{\varphi_{h(e)}\}_{e\in\omega}$*, there is a recursive function* g *such that every* $\varphi_{h(e)}$ *is* g*-honest.* (Blum [1967]) (Hint: let, as usual,

$$h(e,x,z) = \begin{cases} \Phi_{h(e)}(x) & \text{if } \varphi_{h(e)}(x) \simeq z \\ 0 & \text{otherwise} \end{cases}$$

and

$$g(x,z) = \max\{h(e,x,z) : e \le x\}.)$$

b) *The set of* g*-honest functions is r.e.* (Meyer and McCreight [1969]) (Hint: let $\varphi_{h(e,a,b)}$ be defined in such a way as to mimic φ_e when this appears to be g-honest, and interpret the parameters a and b as bounds of, respectively, the number of exceptions to g-honesty and the complexity needed to compute the convergent values of the function for these exceptional values. Thus, if

$$(\forall_\infty x)[\Phi_e(x) \le g(x,\varphi_e(x))]$$

then there are a and b such that $\varphi_e \simeq \varphi_{h(e,a,b)}$. Conversely, every $\varphi_{h(e,a,b)}$ is g-honest.)

c) *Every* g*-honest function has a recursive graph.* (Hint: see VII.2.10.b.)

d) *Any r.e. set of functions* $\{\varphi_{h(e)}\}_{e\in\omega}$ *such that, for every* e *and* x*,*

$$\Phi_{h(e)}(x) \le g(x,\varphi_{h(e)}(x))$$

is measured. (Hint: see VII.2.10.b.)

The exercises show that *the concepts of measured set and of honesty are closely related*. They are not exactly equal, since in a) the g-honest functions might include some functions not in the measured set, and in d) we had to request the honesty condition on every x, and not only for almost every x, to ensure the necessary uniformity.

We now prove the existence of classes of functions with an associated notion of best complexity.

Proposition VII.2.15 *For every measured set of functions* $\mathcal{M}$ *there is a recursive function* g *such that every function* φ *in* $\mathcal{M}$ *has a best complexity modulo* g*. Precisely, for some index* i *of* φ

$$(\forall e)[\varphi \simeq \varphi_e \;\Rightarrow\; (\forall_\infty x)(\Phi_i(x) \le g(x,\Phi_e(x)))].$$

Proof. This is simply a consequence of the fact that the values of a recursive function are never much larger than *any* of its complexities, while for measured (and hence honest) sets, the opposite also holds, i.e. there is *some* complexity around the values.

By VII.2.11, there is a recursive function g_1 such that

$$(\forall e)(\forall_\infty x)[\varphi_e(x) \le g_1(x,\Phi_e(x))]. \qquad \text{(VII.1)}$$

Since $\mathcal{M}$ is measured, by VII.2.14.a there is a recursive function g_2 such that, for every $\varphi \in \mathcal{M}$, there is an index i of it such that

$$(\forall_\infty x)[\Phi_i(x) \leq g_2(x, \varphi_i(x))]. \tag{VII.2}$$

We may suppose that g_2 is monotone in the second argument (otherwise it is enough to consider $\max_{y \leq z} g_2(x,y)$ in place of $g_2(x,z)$).

Given $\varphi_e \in \mathcal{M}$, from VII.1 we have, for almost every x,

$$\varphi_e(x) \leq g_1(x, \Phi_e(x))$$

and, by monotonicity of g_2, for almost every x

$$g_2(x, \varphi_e(x)) \leq g_2(x, g_1(x, \Phi_e(x))).$$

Now choose i such that $\varphi_i \simeq \varphi_e$ and VII.2 holds. Then, for almost every x,

$$\Phi_i(x) \leq g_2(x, \varphi_i(x)) = g_2(x, \varphi_e(x)) \leq g_2(x, g_1(x, \Phi_e(x))).$$

By letting

$$g(x,z) = g_2(x, g_1(x,z))$$

we then have, for almost every x,

$$\Phi_i(x) \leq g(x, \Phi_e(x)). \quad \Box$$

The result just proved has a number of interesting applications:

- *single functions*
  Since a set consisting of a single function with a recursive graph is measured, *every partial recursive function with a recursive graph (in particular, every total recursive function) has a best complexity modulo some recursive function.* The proof shows that *any* index of a partial recursive function with a recursive graph has best complexity modulo some recursive function.

  From the existence of speedable sets (see IX.4.1), proved in Section IX.4, it follows that *not every partial recursive function has a best complexity modulo some recursive function* (see IX.4.2.c).

- *classes of functions*
  Since an r.e. set of total recursive functions is measured, *for any r.e. set of total recursive functions, there is a recursive function g such that every function in the class has a best complexity modulo g.* Examples are the classes of polynomial time computable functions (Section VIII.2), elementary functions (Section VIII.7), primitive recursive functions (Section VIII.8) and, more generally, all classes considered in Chapter VIII. Clearly, the bigger the class, the larger the modulus.

  From the Speed-Up Theorem VII.2.16, it follows that *there is no fixed modulus that works for the class of all (0,1-valued) recursive functions.*

The Speed-Up Theorem

We turn now to the proof of the result just quoted, which is one of the pearls of Complexity Theory.

The statement consists of two complementary parts. The first tells us that, given g, some functions are so difficult to compute that they are insensitive to improvements in their programs by the factor g. For example, the only functions whose square root is bounded by a polynomial are those that are already bounded by a polynomial, and thus the programs of a function that is not computable in polynomial time are insensitive to quadratic improvements in their running time. More generally, the construction of a function insensitive to improvements by the factor g is a simple consequence of Rabin's Theorem VII.2.3, and consists of finding a recursive function with all finite iterations of g as lower bounds of its complexities.

The second part of the statement tells us that, given g, there are functions whose programs are not only insensitive *in principle* to improvements by the factor g, but are *actually* improvable by such a factor. This requires a method to improve given programs, which is the novelty of the next proof.

We turn now to a formal treatment.

Theorem VII.2.16 Blum's Speed-Up Theorem (Blum [1967]) *For any recursive function g, there is a recursive 0,1-valued function f that does not have a best complexity modulo g. Precisely, for every index e of f there is another index i of f such that*

$$(\forall_\infty x)[g(x, \Phi_i(x)) \leq \Phi_e(x)].$$

Proof. The idea of the proof is the following. We first define a decreasing sequence $\{h_e\}_{e\in\omega}$ of uniformly recursive functions, each above the next one by at least a factor g (of course, this can hold only almost everywhere):

1. $(\forall e)(\forall_\infty x)[g(x, h_{e+1}(x)) \leq h_e(x)]$.

We then define, by the methods of VII.2.3, a recursive function f such that, if e is an index for f, then its complexity is above h_e:

2. $(\forall e)[f \simeq \varphi_e \Rightarrow (\forall_\infty x)(h_e(x) \leq \Phi_e(x))]$.

Then we prove that, for each e, there is an index i of f whose complexity is below h_{e+1}:

3. $(\forall e)(\exists i)[f \simeq \varphi_i \wedge (\forall_\infty x)(\Phi_i(x) \leq h_{e+1}(x))]$.

The theorem easily follows, if we suppose that g is monotone in the second argument (otherwise, it is enough to consider $\max_{y\leq z} g(x,y)$ in place of $g(x,z)$).

Indeed, given an index e of f, there is another index i such that 3 holds, i.e. for almost every x

$$\Phi_i(x) \leq h_{e+1}(x).$$

Then, for almost every x,

$$g(x, \Phi_i(x)) \leq g(x, h_{e+1}(x)) \leq h_e(x) \leq \Phi_e(x)$$

by monotonicity of g and conditions 1 and 2.

Condition 1 can be satisfied quite easily. Consider first the case of a unary g: then it would be enough to consider the constant functions with values $g^{(x)}(0)$, and to let h_0 be the diagonal function

$$h_0(x) = g^{(x)}(0),$$

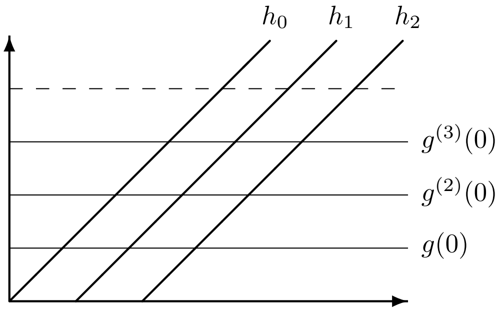
h_1 be the function (obtained by shifting h_0 one step to the right)

$$h_1(x) = g^{(x-1)}(0)$$

and, in general,

$$h_e(x) = g^{(x-e)}(0).$$

Then, after $x = e$ (the point at which h_e starts its ascent from 0), h_e will always be above h_{e+1} by the factor g (since its values are obtained by iterating g one more time than h_{e+1} does).



Similarly, for a binary g, we let

$$h_e(x) = \underbrace{g(x, g(x, g(x, \ldots, g}_{x-e \text{ times}}(x, 0) \cdots))).$$

Then for $x > e$, i.e. for $x - e > 0$,

$$h_e(x) = g(x, \underbrace{g(x, g(x, \ldots, g}_{x-e-1 \text{ times}}(x, 0) \cdots))) = g(x, h_{e+1}(x)).$$

Condition 2 is satisfied as in Rabin's Theorem. The construction of f is as follows. At stage x, we define $f(x)$. Consider e such that:

- $e \leq x$
- e has not yet been cancelled
- $\Phi_e(x) < h_e(x)$.

If there is such an e, choose the least one e_x, define

$$f(x) = 1 - \varphi_{e_x}(x)$$

and cancel e_x. If there is no such e, let

$$f(x) = 0.$$

The proof that 2 is satisfied is exactly as in VII.2.3. The only point to argue about is that f is recursive, since we use a different bound h_e for each e; but the sequence $\{h(e)\}_{e \in \omega}$ is *uniformly* recursive by definition, and thus e appears only as a parameter.

Condition 3 is satisfied by finding quick ways to compute f. The idea is as follows: given u, we can attempt to compute $f(x)$ faster than in the original program given by the definition of f by looking, for $x \geq u$, only at indices e such that $u \leq e \leq x$. Since u is fixed, this procedure works for almost every x, but it does not produce exactly f. To correct the error caused by working only above u, we introduce a parameter v that codes the information needed to recover the original f. Thus, given u, we define v as any sequence number such that:

- $ln(v)$ is even, and $\frac{1}{2} \cdot ln(v)$ gives an upper bound of the arguments where the original definition of f cancels a number less than u
- for $x < \frac{1}{2} \cdot ln(v)$, $(v)_{2x+1}$ gives the original value $f(x)$, and $(v)_{2x+2}$ tells us whether an index was cancelled in the original construction of $f(x)$ and, if so, which one (this is achieved by letting $(v)_{2x+2}$ be 0 if nothing was cancelled, and 1 plus the cancelled index otherwise).

We now define a total function $\varphi \simeq \varphi_{t(u,v)}$ depending recursively and uniformly on the parameters u and v, and reproducing the definition of f when the parameters are interpreted as described above. Suppose that, for $z < x$, we know by induction both $\varphi(z)$ and which indices have been cancelled in the definition of $\varphi(z)$, if any.

If $x < \frac{1}{2} \cdot ln(v)$, then we let

$$\varphi(z) = (v)_{2x+1},$$

and if $(v)_{2x+2} > 0$, then we consider $(v)_{2x+2} - 1$ as cancelled at this stage.
If $x \geq \frac{1}{2} \cdot ln(v)$, then consider e such that:

- $u \leq e \leq x$
- e has not yet been cancelled
- $\Phi_e(x) < h_e(x)$.

If there is such an e, choose the least one e_x, define

$$\varphi(x) = 1 - \varphi_{e_x}(x)$$

and cancel e_x. If there is no such e, let

$$\varphi(x) = 0.$$

Notice that, as argued informally above, for every u, there is v such that

$$\varphi_{t(u,v)} \simeq \varphi_{t(0,0)} \simeq f.$$

To show that 3 holds, we have to look at the definition of $\varphi_{t(u,v)}$. When $x \geq \frac{1}{2} \cdot ln(v)$ we define $\varphi_{t(u,v)}(x)$ by looking at the indices e such that $u \leq e \leq x$, and checking whether $\Phi_e(x) < h_e(x)$. Since $e \geq u$, $h_e(x) \leq h_u(x)$ for almost every x (if $g(x,y) \geq y$, which we can always suppose). If we could conclude from this that

$$(\forall_\infty x)[\Phi_{t(u,v)}(x) \leq h_u(x)], \tag{VII.3}$$

then we would be done: by choosing v such that

$$\varphi_{t(e+1,v)} \simeq \varphi_{t(0,0)} \simeq f$$

we would then have

$$(\forall_\infty x)[\Phi_{t(e+1,v)}(x) \leq h_{e+1}(x)],$$

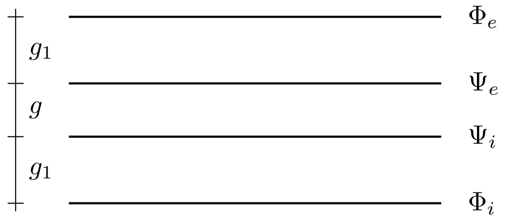
as required by condition 3, $t(e+1, v)$ being the required i.

But VII.3 follows only for particular measures, for example those (such as space complexity) for which the parallel computation property (see p. 22) holds. Then *a single* check that $\Phi_e(x) < h_e(x)$ requires only $h_e(x) \leq h_u(x)$ resource, and by the parallel computation property *all* checks require only $h_u(x)$ resource.

To obtain the result in full generality, we have two possible methods.

- *first (nonuniform) proof*
  We can prove that the Speed-Up Theorem is measure-independent. Then a proof for one particular measure $\{\Phi_e\}_{e\in\omega}$, as done above, is enough.

  Measure-independence is just a trivial observation. Suppose $\{\Psi_e\}_{e\in\omega}$ is another measure, related to $\{\Phi_e\}_{e\in\omega}$ by g_1. If we want to have a function with no best complexity modulo g w.r.t. $\{\Psi_e\}_{e\in\omega}$, it is enough to find a function with no best complexity modulo $g_1 \circ g \circ g_1$ w.r.t. $\{\Phi_e\}_{e\in\omega}$. Indeed, Φ_e can at most be above Ψ_e by a factor g_1, and Φ_i can at most be below Ψ_i by a factor g_1; thus, if Φ_i is below Φ_e by at least $g_1 \circ g \circ g_1$, Ψ_i is below Ψ_e by at least g, as required.



- *second (uniform) proof*
  We can modify the proof given above for a particular measure by turning it upside-down. Instead of *defining* the sequence $\{h_e\}_{e\in\omega}$ ahead of time, we can *find* an appropriate sequence at the end, depending on the measure. In other words, we look for $\{h_e\}_{e\in\omega}$ such that for every e

$$f \simeq \varphi_e \;\Rightarrow\; (\forall_\infty x)[h_e(x) \leq \Phi_e(x)],$$

  and for every index e of f there is another index i of f such that

$$(\forall_\infty x)[g(x, \Phi_i(x)) \leq h_e(x)].$$

  Thus conditions 1 and 3 are now combined into a single condition.

  Since the sequence $\{h_e\}_{e\in\omega}$ was uniformly recursive, we actually had a total recursive function h such that $h(e,x) = h_e(x)$. By definition,

$$h(e,x) \simeq \begin{cases} 0 & \text{if } x \leq e \\ g(x, h(e+1,x)) & \text{if } x > e. \end{cases}$$

  We can reproduce the construction of f by using any total recursive function φ_j in place of the particular h used above, thus obtaining a function depending on the parameter j. We now want to *find* j such that for every index e of f thus obtained, there is another index i of f such that

$$\varphi_j(e,x) = \begin{cases} 0 & \text{if } x \leq e \\ g(x, \Phi_i(x)) & \text{if } x > e. \end{cases}$$

The proof above suggests that we define a function depending on the additional parameters u and v, thus obtaining $\varphi_{t(u,v,j)}$, and that given e, the required i will be $t(e+1,v,j)$ for some appropriate v, namely one such that

$$\varphi_{t(e+1,v,j)} \simeq \varphi_{t(0,0,j)}.$$

We are thus looking for j such that

$$\varphi_j(e,x) = \begin{cases} 0 & \text{if } x \leq e \\ g(x, \Phi_{t(e+1,v,j)}(x)) & \text{if } x > e. \end{cases}$$

This does not define a function yet, since the right hand side still depends on v. This dependency can however be easily avoided, for example by letting

$$\varphi_j(e,x) = \begin{cases} 0 & \text{if } x \leq e \\ \max_{v \leq x} g(x, \Phi_{t(e+1,v,j)}(x)) & \text{if } x > e, \end{cases}$$

Such a j exists by the Fixed-Point Theorem, and we only have to argue that the φ_j thus obtained is total.

To compute $\varphi_{t(e+1,v,j)}(x)$ for any v, one looks only at indices e' between $e+1$ and x, and checks whether

$$\Phi_{e'}(x) < h_{e'}(x) = \varphi_j(e',x).$$

This requires that $\varphi_j(e',x)$ is defined. But if $e+1 \leq e' \leq x$, then $x - e' < x - e$. Thus φ_j is total by induction on $x - e$ (the base case being defined, since $\varphi_j(e,x) = 0$ if $x - e = 0$, i.e. if $x \leq e$). $\square$

The Speed-Up Theorem has a number of interesting consequences. First, *there is no notion of best complexity for all (0,1-valued) total recursive functions, even if one requires optimality (modulo some fixed recursive function) only on infinitely many arguments*. This is a consequence of the fact that the proof produces a speed-up on almost all arguments, and not only on infinitely many (as would be enough to rule out a notion of best complexity).

Second, we can now prove that *no computer can be optimal for every purpose*: no matter how good a computer is, there are always functions on which such a computer behaves very badly. For example, let M_g and M_b be two Turing machines computing the universal function $\varphi(e,x) \simeq \varphi_e(x)$, of which the former is very good (e.g. it performs a million steps per second) while the latter is hopelessly bad (e.g. it performs one step every million years). Let $\{\Phi_e\}_{e\in\omega}$ and $\{\Psi_e\}_{e\in\omega}$ be the time complexity measures relative to M_g and M_b, respectively. By recursive relatedness VII.1.15, there is a recursive function g such that, for every i and almost every x, if $\varphi_i(x)\downarrow$, then

$$\Psi_i(x) \leq g(x, \Phi_i(x)).$$

By the Speed-Up Theorem relative to g (plus 1), there is a recursive function f such that, for every index e of f, there is another index i of f such that

$$(\forall_\infty x)[g(x, \Phi_i(x)) < \Phi_e(x)].$$

Thus

$$(\forall_\infty x)[\Psi_i(x) < \Phi_e(x)].$$

In other words, no matter how fast the good computer M_g computes f, there is a faster way of computing it on the bad computer M_b!

The Speed-Up Theorem is most useful in questions of philosophical hygiene, such as those just discussed. On the other hand, *its practical interest is quite limited*. First, the result is existential and it only asserts that *some* artificially concocted functions have speed-ups (see VII.2.17.d, VII.4.8 and VII.4.9 for universal results). Second, the better programs cannot be obtained effectively from the given ones (Blum [1971], see VII.2.17.b), and in any case they are better only on almost all arguments (in particular, the exceptions could include all arguments of specific interest in any particular case).

Obviously, *by repeated speed-ups, both the number of exceptions and the size of the better programs become larger*. Otherwise, one would obtain infinitely descending sequences (of complexities on a given argument in the first case, and of sizes in the second). The sizes of the faster programs can sometimes (Blum [1971], see VII.2.17.c) but not always (Helm and Young [1971], Meyer and Fischer [1972]) be bounded recursively in the sizes of the given programs. Also, the exceptions cannot be bounded recursively in the sizes of the given programs (Schnorr [1973], Kummer [1993], Bridges and Calude [1994]).

Exercises VII.2.17 a) *The function of the Speed-Up Theorem can be arbitrarily complicated.* (Hint: given t recursive, use the method of VII.2.3 in the proof of VII.2.16, to ensure that no complexity of f is below t infinitely often.)

b) *The Speed-Up Theorem does not hold constructively, in the sense that if g is sufficiently large, then there are no recursive functions f and h such that*

$$f \simeq \varphi_e \ \Rightarrow\ f \simeq \varphi_{h(e)} \ \wedge\ (\forall_\infty x)[g(x, \Phi_{h(e)}(x)) \leq \Phi_e(x)].$$

(Blum [1967], Schnorr [1973]) (Hint: given g, f and h, we find an index e of f such that

$$f \simeq \varphi_e \simeq \varphi_{h(e)} \ \wedge\ (\exists_\infty x)[\Phi_e(x) < g(x, \Phi_{h(e)}(x))].$$

We use the Double Fixed-Point Theorem (see II.2.11.b) to define two indices a and b, one of which will actually work.

We fix an index i for f, let $x = 0$ and see what happens first:

1. if $\varphi_i(x)\downarrow$, i.e. $\Phi_i(x) \leq \Phi_{h(a)}(x), \Phi_{h(b)}(x)$, let $\varphi_a(x) \simeq \varphi_b(x) \simeq f(x)$;
2. if $\varphi_{h(a)}(x)\downarrow$, i.e. $\Phi_{h(a)}(x) \leq \Phi_{h(b)}(x)$, let $\varphi_a(x) \simeq f(x)$ and $\varphi_b(x) \simeq \varphi_{h(a)}(x)$;
3. if $\varphi_{h(b)}(x)\downarrow$, i.e. $\Phi_{h(b)}(x) \leq \Phi_{h(a)}(x)$, let $\varphi_b(x) \simeq f(x)$ and $\varphi_a(x) \simeq \varphi_{h(b)}(x)$.

Then we continue by letting $\varphi_a(n) \simeq \varphi_b(n) \simeq f(n)$, unless we can check in less than n steps that

$$\varphi_{h(a)}(x) \simeq \varphi_{h(b)}(x) \simeq f(x),$$

in which case we let $x = n$ and start again.

Notice that it can never be the case that the check actually fails. Otherwise, suppose x is the smallest element for which this happens. Then at least one of $\varphi_{h(a)}(x)$ and $\varphi_{h(b)}(x)$ is different from $f(x)$, e.g. $\varphi_{h(a)}(x)$. But then the construction produces $\varphi_a \simeq f$ and, by the assumption on h, $\varphi_{h(a)} \simeq f$ too, contradiction.

Since such checks never fail, the construction produces infinitely many x's as above. Then $\varphi_a \simeq f$, because either φ_a is directly defined to be equal to f, or it is defined to be equal to $\varphi_{h(b)}$ on inputs on which the latter agrees with f (because the check never fails). Similarly, $\varphi_b \simeq f$. Then $\varphi_{h(a)} \simeq \varphi_{h(b)} \simeq f$ too, by the assumption on h.

If the first case of the construction was used for infinitely many such x's, then both a and b would work. For example, $\varphi_a(x)$ was defined as $\varphi_i(x)$, and $\Phi_i(x) \leq \Phi_{h(a)}(x)$. Then $\Phi_a(x)$ is not much bigger than $\Phi_{h(a)}(x)$ (one cannot assert that it is actually smaller because the full definition of φ_a does not simply copy φ_i down, but the complexities of φ_a and φ_i are related by a fixed recursive function, which is bounded by g if the latter is sufficiently large).

If the first case of the construction was used only for finitely many such x's, then the remaining two cases were used infinitely often. If the second case was used infinitely often then φ_b would work, because $\varphi_b(x)$ was defined as $\varphi_{h(a)}(x)$, and $\Phi_{h(a)}(x) \leq \Phi_{h(b)}(x)$. Then $\Phi_b(x)$ is not much bigger than $\Phi_{h(b)}(x)$. Similarly for the third case, where φ_a would work.)

c) *For some function with speed-up, the size of the better program can be bounded recursively in the size of the given one.* (Blum [1971]) (Hint: given u, in VII.2.16 we obtained a faster program for f by considering only indices $e \geq u$. To be able to do this, we needed to know which indices below u were cancelled in the original definition of f, and there are only finitely many possibilities. To each one corresponds a program whose size can be obtained effectively, and one of these is the right one for f, i.e. the one in which the guess about which indices were cancelled is the right one. We obtain the bound by recalling that an index faster than e was obtained by considering $u = e + 1$.)

d) *For any recursive function g there is a measure $\{\Phi_e\}_{e\in\omega}$ such that no recursive function has a best program modulo g.* (Hint: we may suppose $g(x, y) \geq y$, by possibly considering $g(x, y) + y$. Let $\{\Psi_e\}_{e\in\omega}$ be a measure for the acceptable system $\{\psi_e\}_{e\in\omega}$, and define an acceptable system $\{\varphi_e\}_{e\in\omega}$ and a measure $\{\Phi_e\}_{e\in\omega}$ for it by letting

$$\varphi_{\langle e,n\rangle} \simeq \psi_e \qquad \text{and} \qquad \Phi_{\langle e,n\rangle}(x) = \underbrace{g(x, g(x, g(x, \ldots, g}_{x-n \text{ times}}(x, \Psi_e(x))\cdots))).$$

For $x > n$, one then has

$$g(x, \Phi_{\langle e,n+1\rangle}(x)) = \Phi_{\langle e,n\rangle}(x),$$

and thus ψ_e has no best program modulo g.)

There are various *extensions* of the Speed-Up Theorem.

- Alton [1976] shows that the structure of functions with speed-ups is very rich (the ordering by inclusion of their complexity classes embeds every countable partial ordering). In the same direction, Calude, Istrate and Zimand [1992], Calude and Zimand [1996] show that the class of functions with speed-ups is topologically large.

- Meyer and Fischer [1972] show that the theorem holds not only for recursive functions g, but also for total recursive operators G, i.e. there is a 0,1-valued recursive function f that does not have a best complexity modulo G. More precisely, for every index e of f there is another index i of f such that
$$(\forall_\infty x)[G(\Phi_i)(x) \leq \Phi_e(x)]$$
(see Young [1973] and Calude [1988] for proofs).

- Young [1971b] and Dawes [1982] show that some sets are so difficult to enumerate that significant speed-ups in their enumerations can be obtained only by changing the order in which the sets are enumerated, and not merely by speeding-up the computation of the enumerating function. More precisely, there are infinite sets such that, given any enumeration of them, there is another enumeration and a program for it that runs faster than any program for the first enumeration.

- Constable and Borodin [1972] and Alton [1980] have proved speed-up theorems for subclasses of recursive functions.

By looking at the proof of VII.2.16, Meyer and Fischer [1972] suggested an extension of the notion of best complexity using not only a single program, but a *complexity sequence of good programs* (cofinal in the complexities of a given function). This idea has its best applications with partial functions, see IX.4.21 and IX.4.24. For total functions, Levin [1973], [1974], and Meyer and Winklmann [1979] characterize the sequences of programs that are complexity sequences of (0,1-valued) recursive functions. See also Van Emde Boas [1975], Schnorr and Stumpf [1975] and Lynch [1975].

VII.3 Complexity Classes

The emphasis in the last section was on single functions and their complexities. Here we enlarge our point of view and consider classes of recursive functions determined by complexity bounds. This study provides an abstract foundation to the theory of subrecursive classes, which is developed in detail in Chapter VIII.

Definition VII.3.1 *Given a complexity measure $\{\Phi_e\}_{e\in\omega}$ and a recursive function t, we call a* **complexity class with name t** *the set $\mathcal{C}_t^\Phi$ of recursive functions that admit t as an upper bound of some of their complexities:*

$$\mathcal{C}_t^\Phi = \{f : (\exists e)[f \simeq \varphi_e \ \wedge \ (\forall_\infty x)(\Phi_e(x) \leq t(x))]\}.$$

When no confusion arises we drop reference to the measure $\{\Phi_e\}_{e\in\omega}$, and simply write $\mathcal{C}_t$.

Although we have restricted our definition to complexity classes of total recursive functions, *complexity classes of partial recursive functions* have also been considered in the literature. See Meyer and McCreight [1969], Landweber and Robertson [1972], Robertson [1974], Moll and Meyer [1974], Gill and Blum [1974], Barashko and Roizen [1976], and Barashko [1977].

In the opposite direction, one can restrict the notion of a complexity class by considering only formally provable properties. See Young [1977], Hartmanis [1978a], [1979], and Joseph and Young [1981].

Exercises VII.3.2 Structural properties of complexity classes.

a) *Any complexity class $\mathcal{C}_t$ closed under finite variants is r.e.* (Borodin [1972]) (Hint: fix $h \in \mathcal{C}_t$ and define

$$\varphi_{g(e,a,b)}(x) \simeq \begin{cases} \varphi_e(x) & \text{if } (\forall z \leq a)(\Phi_e(z) \leq b) \wedge (\forall z)_{a<z\leq x}(\Phi_e(z) \leq t(z)) \\ h(x) & \text{otherwise.} \end{cases}$$

If $f \in \mathcal{C}_t$ then for some e, a and b is $f \simeq \varphi_{g(e,a,b)}$. Conversely, $\varphi_{g(e,a,b)} \in \mathcal{C}_t$ because either the first clause holds for infinitely many x, and then $\varphi_{g(e,a,b)} \simeq \varphi_e \in \mathcal{C}_t$, or $\varphi_{g(e,a,b)}$ is almost equal to h, and hence in $\mathcal{C}_t$ by hypothesis.)

b) *Any complexity class containing all finite variants of a single function is r.e.* (Meyer and McCreight [1969]) (Hint: by the proof of part a).)

c) *Every measure admits r.e. complexity classes.* (Hint: by part b), since the set of functions equal to 0 almost everywhere is r.e., and thus there is a recursive function t majorizing at least one complexity for each function in the set.)

d) *There are measures for which every complexity class is r.e.* (Hartmanis and Stearns [1965]) (Hint: apply part a) to space complexity.)

e) *There are measures for which not every complexity class is r.e.* (Lewis [1971a], Landweber and Robertson [1972]) (Hint: let f be an increasing recursive function such that $\varphi_{f(z)}$ is the constant function with value z. Define a measure as follows:

$$\Phi_e(x) \simeq \begin{cases} 0 & \text{if } e = f(z) \wedge z \notin \mathcal{K}_x \\ 1 & \text{if } e = f(z) \wedge z \in \mathcal{K}_x \\ \Psi_e(x) + 2 & \text{if } (\forall z)(e \neq f(z)), \end{cases}$$

where $\{\Psi_e\}_{e\in\omega}$ is any measure. Let t be the constant function with value 0. A function is in $\mathcal{C}_t^\Phi$ if and only if, for some z, it is the constant function with value z and $z \notin \mathcal{K}$. Thus, if $\varphi_{h(x)}$ enumerates $\mathcal{C}_t^\Phi$, then $\varphi_{h(x)}(0)$ enumerates $\overline{\mathcal{K}}$, which is a contradiction.)

f) *Every complexity class $\mathcal{C}_t$ is co-r.e.* (Landweber and Robertson [1972]) (Hint: let E_i be an infinite recursive set of indices for φ_i, such that $E = \bigcup_{i\in\omega} E_i$ is recursive. Let $E_i = \{e_0^i < e_1^i < \cdots\}$, and define

$$e_n^i \in A \Leftrightarrow (\exists x)[\Phi_i(x) > t(x) + (n - x)].$$

Then A is r.e. and

$$E_i - A \neq \emptyset \Leftrightarrow \varphi_i \text{ is total } \wedge (\forall_\infty x)(\Phi_i(x) \leq t(x)).$$

Thus $\bigcup_{i\in\omega}(E_i - A) = E - A$ is a co-r.e. set of indices for $\mathcal{C}_t$.)

g) *Complexity classes are small, in the sense that it is possible to build effectively a recursive function not in the class.* This is a kind of effective meagerness, defined precisely in Calude [1982]. (Hint: each complexity class is contained in an r.e. complexity class since, by part b), it is enough to add to it all finite modifications of a single function, e.g. of the constant function 0. And it is always possible to diagonalize against r.e. classes of recursive functions.)

h) *If $\mathcal{C}$ is an r.e. class of recursive functions that contains all finite variants of a single function, then $\theta\mathcal{C} = \{e : \varphi_e \in \mathcal{C}\}$ is independent of $\mathcal{C}$ (up to recursive isomorphism). Moreover, $\theta\mathcal{C} \in \Delta_3^0 - (\Sigma_2^0 \cup \Pi_2^0)$.* (Lewis [1971]) (Hint: we use the sets $\mathbf{Fin} \in \Sigma_2^0 - \Pi_2^0$ and $\mathbf{Tot} \in \Pi_2^0 - \Sigma_2^0$ defined in X.9.6. Let

$$\begin{aligned} e \in A \quad &\Leftrightarrow \quad \varphi_{h(e)} \text{ is total and bounded} \\ &\Leftrightarrow \quad \varphi_e \in \mathbf{Tot} \wedge (\exists n)(\forall x)(\varphi_e(x) \leq n). \end{aligned}$$

Then $A \in \Delta_3^0$ since $\mathbf{Tot} \in \Pi_2^0$. $\mathbf{Tot} \leq_1 A$ via the function $\varphi_{f(e)} \simeq 0 \cdot \varphi_e$. $\mathbf{Fin} \leq_1 A$ via the function $\varphi_{g(e)}(x)$ which gives the number of elements of $\mathcal{W}_{e,x}$.

To show $\theta\mathcal{C} \leq_1 A$, let B be an r.e. set such that

$$e \in \theta\mathcal{C} \Leftrightarrow (\exists x)(x \in B \wedge \varphi_e \simeq \varphi_x),$$

which exists because $\mathcal{C}$ is r.e. If

$$\varphi_{h(e)}(x) \simeq \begin{cases} \mu z\,(z \text{ is generated in } B \wedge \varphi_z \text{ and } \varphi_e \text{ agree up to } x) & \text{if } z \text{ exists} \\ \text{undefined} & \text{otherwise,} \end{cases}$$

then

$$e \in \theta\mathcal{C} \Leftrightarrow \varphi_{h(e)} \text{ is total and bounded.}$$

To show $A \leq_1 \theta\mathcal{C}$, suppose $\mathcal{C}$ contains all finite variants of f. Let g be a recursive function majorizing every function in $\mathcal{C}$, which exists because $\mathcal{C}$ is r.e. If

$$\varphi_{t(e)}(x) \simeq \begin{cases} f(x) & \text{if } \varphi_e(x) \leq \max\{\varphi_e(0), \dots, \varphi_e(x-1)\} \\ g(x) & \text{if } \varphi_e(x) > \max\{\varphi_e(0), \dots, \varphi_e(x-1)\} \\ \text{undefined} & \text{if } \varphi_e(y)\uparrow, \text{ for some } y \leq x, \end{cases}$$

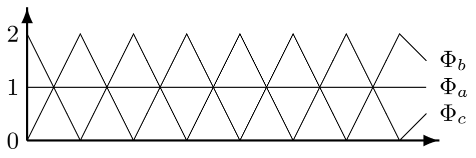
then $\varphi_{t(e)}$ is total if and only if φ_e is total. If φ_e is unbounded, then $\varphi_{t(e)}$ agrees with g infinitely often and hence is not in $\mathcal{C}$. If φ_e is bounded, then $\varphi_{t(e)}$ agrees with f almost everywhere and hence is in $\mathcal{C}$.)

Exercises VII.3.3 Set-theoretical operations on complexity classes.

a) *There are measures whose complexity classes are not closed under union.* (Hint: we want to find a measure $\{\Phi_e\}_{e\in\omega}$ and two recursive functions t_1 and t_2 such that for no recursive function t is $\mathcal{C}_t^\Phi = \mathcal{C}_{t_1}^\Phi \cup \mathcal{C}_{t_2}^\Phi$. Let $\{\Psi_e\}_{e\in\omega}$ be any measure, and define

$$\Phi_e(x) \simeq \begin{cases} \Psi_e(x) + 3 & \text{if } e \neq a, b, c \\ 1 & \text{if } e = a \\ 0 & \text{if } (e = b \wedge x \text{ even}) \vee (e = c \wedge x \text{ odd}) \\ 2 & \text{if } (e = c \wedge x \text{ even}) \vee (e = b \wedge x \text{ odd}), \end{cases}$$

where $f \simeq \varphi_a$, $g \simeq \varphi_b$ and $h \simeq \varphi_c$ are three different recursive functions.



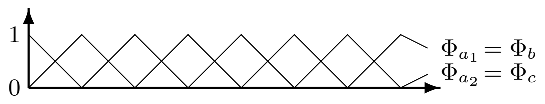
Let $t_1 = \Phi_b$ and $t_2 = \Phi_c$. Then $\mathcal{C}_{t_1}^\Phi = \{g\}$ and $\mathcal{C}_{t_2}^\Phi = \{h\}$, but for no t is $\mathcal{C}_t^\Phi = \{g, h\}$ because either infinitely many points of g or h are excluded, or f is also included.)

b) *For every measure the complexity classes are not closed under union, and hence are not linearly ordered by inclusion.* (Hartmanis and Stearns [1965], Meyer and McCreight [1969]) (Hint: modify the idea of part a) by choosing t_1 large enough, and t_2 oscillating w.r.t. t_1 and such that every function that is above both also bounds some function not in $\mathcal{C}_{t_1} \cup \mathcal{C}_{t_2}$.)

c) *There are measures whose complexity classes are not closed under intersection.* (Landweber and Robertson [1972]) (Hint: let $\{\Psi_e\}_{e\in\omega}$ be any measure, and define

$$\Phi_e(x) \simeq \begin{cases} \Psi_e(x) + 2 & \text{if } e \neq a_1, a_2, b, c \\ 0 & \text{if } (e = a_1 \text{ or } b \wedge x \text{ even}) \vee (e = a_2 \text{ or } c \wedge x \text{ odd}) \\ 1 & \text{if } (e = a_2 \text{ or } c \wedge x \text{ even}) \vee (e = a_1 \text{ or } b \wedge x \text{ odd}), \end{cases}$$

where $f \simeq \varphi_{a_1} \simeq \varphi_{a_2}$, $g \simeq \varphi_b$ and $h \simeq \varphi_c$ are three different recursive functions.



Let $t_1 = \Phi_b$ and $t_2 = \Phi_c$. Then $\mathcal{C}^{\Phi}_{t_1} = \{f, g\}$ and $\mathcal{C}^{\Phi}_{t_2} = \{f, h\}$, but for no t is $\mathcal{C}^{\Phi}_{t} = \{f\}$ because either infinitely many points of f are excluded, or one of g and h is included.)

d) *There are measures whose complexity classes are closed under intersection.* (Hartmanis and Stearns [1965]) (Hint: any measure with the parallel computation property, see p. 22.)

Borodin, Constable and Hopcroft [1969], Meyer and McCreight [1969], Young [1971a], and Moll [1976] have studied the structure of complexity classes under inclusion. Some properties hold for every measure (for example, embeddability of every countable partial ordering), while others depend on the measure (for example, density).

Hierarchies for the recursive functions: successor levels ⋆

Complexity classes relative to a given measure provide a way of classifying recursive functions according to their complexity, and can be used in the construction of hierarchies for the recursive functions. Since no complexity class is exhaustive (by Rabin's Theorem VII.2.3), the first problem that arises is how to extend a given class, in a uniform way.

The next result is just a restatement of the Compression Theorem VII.2.8 and tells us that, given a measured set of functions, there is a 'jump operator' that allows us to extend uniformly any complexity class with a name in the measured set.

Proposition VII.3.4 Jump Operators for Measured Sets of Complexity Classes (Blum [1967]) *For any measured set of partial recursive functions $\{\varphi_{h(e)}\}_{e\in\omega}$, there is a recursive function g such that*

$$\varphi_{h(e)} \text{ total} \Rightarrow \mathcal{C}_{\varphi_{h(e)}} \subset \mathcal{C}_{g\circ\varphi_{h(e)}}.$$

Proof. As in the proof of the Compression Theorem (see VII.2.10.a), one can obtain recursive functions f and g such that:

1. $\varphi_{f(e)}$ is total if and only if $\varphi_{h(e)}$ is total
2. $(\forall i)[\varphi_{f(e)} \simeq \varphi_i \Rightarrow (\forall_\infty x)(\varphi_{h(e)}(x) < \Phi_i(x))]$
3. $(\forall_\infty x)[\Phi_{f(e)}(x) \leq g(x, \varphi_{h(e)}(x))]$.

Let e be such that $\varphi_{h(e)}$ is total, and consider $\mathcal{C}_{\varphi_{h(e)}}$. Then $\mathcal{C}_{\varphi_{h(e)}} \subseteq \mathcal{C}_{g\circ\varphi_{h(e)}}$ because, by 2 and 3, for almost every x

$$\varphi_{h(e)} < \Phi_{f(e)}(x) \leq g(x, \varphi_{h(e)}(x)).$$

Moreover, the inclusion is proper because $\varphi_{f(e)}$ is a total recursive function (by 1) which is in $\mathcal{C}_{g\circ\varphi_{h(e)}}$ (by 3) but not in $\mathcal{C}_{\varphi_{h(e)}}$ (by 2). □

At this point, to obtain a jump operator that allows us to extend every complexity class uniformly we still need to find a measured set containing names for every complexity class. This is the content of the next theorem, while the next exercise shows that the most natural measured set would not work in general.

Exercise VII.3.5 *There are measures $\{\Phi_e\}_{e\in\omega}$ for which not every complexity class can be named by a step-counting function*, i.e. for some recursive function t and every e, $\mathcal{C}_t \neq \mathcal{C}_{\Phi_e}$. (Meyer and McCreight [1969]) (Hint: consider any measure such that $\Phi_e \in \mathcal{C}_{\Phi_e}$ for every e, e.g. space complexity. By VII.3.4, there is a monotone recursive function g such that $g(x, y) \geq y$ and $\mathcal{C}_{\Phi_e} \subset \mathcal{C}_{g\circ\Phi_e}$. Fix f recursive and let $t_0 = f$, $t_{n+1} = g \circ t_n$. By the Union Theorem VII.3.7, there is t recursive such that $\mathcal{C}_t = \bigcup_{n\in\omega} \mathcal{C}_{t_n}$. If $\mathcal{C}_t = \mathcal{C}_{\Phi_e}$, then $\Phi_e \in \mathcal{C}_t$, so $\Phi_e(x) \leq t_n(x)$ for some n and almost every x. By monotonicity, $g(x, \Phi_e(x)) \leq t_{n+1}(x)$ for almost every x, and thus

$$\mathcal{C}_t = \mathcal{C}_{\Phi_e} \subset \mathcal{C}_{g\circ\Phi_e} \subseteq \mathcal{C}_{t_{n+1}} \subseteq \mathcal{C}_t,$$

which is a contradiction.)

Theorem VII.3.6 Naming Theorem (Meyer and McCreight [1969]) *For any measure there is a measured set of functions containing a name for every complexity class. Precisely, there is a recursive function h such that:*

1. *the predicate $\varphi_{h(i)}(x) \simeq y$ is recursive (uniformly in i, x and y)*
2. *if φ_i is total, so is $\varphi_{h(i)}$*
3. *if φ_i is total, $\mathcal{C}_{\varphi_i} = \mathcal{C}_{\varphi_{h(i)}}$.*

Proof. h is obtained at the end, by noticing that the proof is uniform in i. Thus we fix i, and just write φ for $\varphi_{h(i)}$.

To satisfy condition 2, we decide to define $\varphi(x)$, if not yet defined, when we discover that $\varphi_i(x)\downarrow$. More precisely, if $\Phi_i(x) \simeq z$, then we define $\varphi(x)$ not later than at stage $\langle x, z\rangle$.

To satisfy condition 1, we decide that if we define $\varphi(x)$ at stage $\langle x, z\rangle$, then $\varphi(x) \geq z$. To check whether $\varphi(x) \simeq y$, we then run the construction through the stages $\langle x, z\rangle$ for $z \leq y$, and see if $\varphi(x)$ has received the value y at some of these stages.

To satisfy condition 3, we have the following requirements when φ_i is total:

- $\mathcal{C}_{\varphi_i} \subseteq \mathcal{C}_{\varphi}$
  This is ensured by satisfying, for every e,

$$H_e \ : \ (\forall_\infty x)(\Phi_e(x) \leq \varphi_i(x)) \ \Rightarrow \ (\forall_\infty x)(\Phi_e(x) \leq \varphi(x)).$$

  These requirements tend to keep φ high, and we satisfy them as follows: any time we find a value x such that $\Phi_e(x) \leq \varphi_i(x)$, we try to have $\varphi(x) \geq \Phi_e(x)$.

- $\mathcal{C}_\varphi \subseteq \mathcal{C}_{\varphi_i}$
  This is a symmetrical condition, but since φ_i is given while φ is built, it is more convenient to look at the contrapositive form. Thus, for each e, we want to satisfy

  $$L_e \ : \ (\exists_\infty x)(\varphi_i(x) < \Phi_e(x)) \ \Rightarrow \ (\exists_\infty x)(\varphi(x) < \Phi_e(x)).$$

  These requirements tend to keep φ low, and we satisfy them as follows: any time we find a value y such that $\varphi_i(y) < \Phi_e(y)$, we assign the index e a mark, and in the following we work to obtain a value x such that $\varphi(x) < \Phi_e(x)$, erasing the mark when this is achieved.

These requirements are infinitary, and to satisfy each of them we have to act infinitely often (unlike the requirements considered in Section 2, that could be satisfied by a single action).

Since the two types of requirements are in conflict, one trying to keep φ low and the other trying to push it high, we order the requirements into a priority list. However, it is not possible to consider the natural ordering by magnitude, because of the following problem.

Suppose e is such that $(\exists_\infty x)(\varphi_i(x) < \Phi_e(x))$. When e is marked, we try to define $\varphi(x)$ sufficiently low for some x, and this might conflict with our desire of satisfying $H_{e'}$ for some $e' > e$. In figurative language, satisfaction of L_e might **injure** $H_{e'}$. But L_e is infinitary, and thus $H_{e'}$ might be injured infinitely often (by defining $\varphi(x)$ too low for infinitely many x), and hence not satisfied (since it requires satisfaction almost everywhere).

A solution to the problem is as follows. At a given stage, we only consider finitely many indices (as usual), with a given priority list. When we satisfy L_e for some marked e, we erase the mark and put e at the end of the list, thus giving all other conditions higher priority. In other words, we have a **dynamic priority list**. The reason why this succeeds is that L_e is not as demanding as H_e: it only requires action infinitely often, not almost everywhere.

Thus, the main feature of our strategy is that at any given stage, we have an ordered set of indices, some of them marked and some not. If at that stage we want to define $\varphi(x)$, we try to have:

- $\varphi(x) < \Phi_e(x)$ if e is marked
- $\varphi(x) \geq \Phi_e(x)$ otherwise.

If some of these conditions conflict, we solve the conflict by turning to the priority list.

The construction is as follows. At step $s+1$, first put index s in the last place of the priority list.

For each $y \leq s$ such that $\Phi_i(y) \leq s$, compute $\varphi_i(y)$ (since we know it is defined). Give any $e \leq s$ such that

$$\varphi_i(y) < \Phi_e(y)$$

a mark, if it does not have one already (thus making sure that e does not receive marks at different times via the same y).

If $s = \langle x, z \rangle$, $\varphi(x)$ has not yet been defined, and $\Phi_i(x) \geq z$ (i.e. $\Phi_i(x) < z$ does not hold), we attempt a definition of $\varphi(x)$. Consider e such that:

- e is marked
- $\Phi_e(x) > z$
- $\Phi_{e'}(x) \leq z$, for every index e' without a mark and with priority higher than e.

If there is such an e, choose the one with the highest priority e_x, define

$$\varphi(x) = \Phi_{e_x}(x) - 1,$$

erase the mark for e_x, and move e_x to the end of the priority list. By doing so we achieve the following:

- we obtain $\varphi(x) < \Phi_{e_x}(x)$
- since $\Phi_{e'}(x) \leq z \leq \varphi(x)$, we do not injure conditions $H_{e'}$ with priority higher than L_{e_x}.

If there is no such e, we go to the next stage, unless $\Phi_i(x) \simeq z$. In this case we want to define $\varphi(x)$ anyway (to satisfy condition 2), and we let

$$\varphi(x) = \max\{\Phi_i(x), \varphi_i(x)\}.$$

The reasons for this are the following:

- by letting $\varphi(x) \geq \Phi_i(x) = z$, we satisfy the strategy for condition 1
- by letting $\varphi(x) \geq \varphi_i(x)$, we do not injure any condition H_e (this is important, since H_e requires satisfaction almost everywhere).

Conditions 1 and 2 are satisfied by construction. It remains to show that condition 3 is also satisfied.

- *L_e is satisfied, for each e*
  Suppose $(\exists_\infty x)(\varphi_i(x) < \Phi_e(x))$. Then index e has a mark infinitely often during the construction. There are two cases.

If for infinitely many times e receives a mark, then for infinitely many times we defined $\varphi(x)$ in the right way, i.e. $\varphi(x) < \Phi_e(x)$ (as we do every time e loses its mark). Thus L_e is satisfied.

If e never loses its mark from a certain stage on, go to a stage where e has already received its permanent mark, and indices with higher priority have a permanent position and situation (i.e. they are not moved after e in the priority list after this stage, and if they receive a permanent mark then they already have it). Since e does not lose its mark afterwards, we are never in the condition of the construction, i.e. for almost every x there is an index e' without a mark and with priority higher than e such that

$$\Phi_e(x) \leq \Phi_{e'}(x) \leq \varphi_i(x),$$

because otherwise e would lose its mark, or e' would obtain one. But this contradicts the hypothesis that $(\exists_\infty x)(\varphi_i(x) < \Phi_e(x))$.

- *H_e is satisfied, for each e*
  Suppose $(\forall_\infty x)(\Phi_e(x) \leq \varphi_i(x))$. Then e obtains a mark only finitely many times, at most once for each y such that $\varphi_i(y) < \Phi_e(y)$. There are two cases.

  If every time e obtains its mark then it also loses it, consider a stage after this has happened for the last time, and after every index with higher priority that is ever moved after e in the priority list has been moved: from then on the construction always ensures that $\Phi_e(x) \leq \varphi(x)$, i.e. $(\forall_\infty x)(\Phi_e(x) \leq \varphi(x))$. Note that here it is crucial to know that once an index loses its mark then it goes to the end of the priority list, and it no longer interferes.

  If e has a permanent mark, as above for almost every x there is an index e' without a mark and with priority higher than e such that

$$\Phi_e(x) \leq \Phi_{e'}(x) \leq \varphi_i(x).$$

  But the construction always ensures $\varphi_i(x) \leq \varphi(x)$ or $\Phi_{e'}(x) \leq \varphi(x)$, and thus $(\forall_\infty x)(\Phi_e(x) \leq \varphi(x))$. $\square$

It is a necessary feature of the proof to work on indices i of functions, rather than on the functions φ_i themselves. Indeed, *the function h cannot be extensional*, in the sense that it cannot satisfy the condition

$$\varphi_i \simeq \varphi_j \;\Rightarrow\; \varphi_{h(i)} \simeq \varphi_{h(j)}.$$

Otherwise we could combine the Naming Theorem with VII.3.4, and obtain a total recursive operator F such that, for every t, $\mathcal{C}_t \subset \mathcal{C}_{F(t)}$ (by first going from t to a measured name for $\mathcal{C}_t$, and then by applying the jump operator relative to the measured set $\{\varphi_{h(i)}\}_{i\in\omega}$), contradicting the extension of the Gap Theorem quoted after VII.3.11.

Extensions of the Naming Theorem (showing, in particular, that a naming procedure can be badly behaved) are given in Werner [1971], Moll and Meyer [1974], and Van Emde Boas [1978].

Hierarchies for the recursive functions: limit levels

The results of the previous subsection show that there is a natural jump operator that allows us to extend uniformly and in a proper way any given complexity class. One can iterate the procedure infinitely often, thus building a hierarchy of complexity classes. Natural and complementary questions then arise. First, is the procedure exhaustive? Second, can it be continued in the transfinite?

The next result shows that under a natural uniformity condition:

- on the negative side, no exhaustive hierarchy can have length ω
- on the positive side, the process of building transfinite hierarchies can always be continued at limit stages.

Recall that
$$(\forall_\infty x)(t(x) \leq t'(x)) \;\Rightarrow\; \mathcal{C}_t \subseteq \mathcal{C}_{t'}.$$
The hypothesis of the next result is stronger, in that it requires that t is below t' for *all* x.

Theorem VII.3.7 Union Theorem (Meyer and McCreight [1969]) *If $\{t_n\}_{n\in\omega}$ is a sequence of uniformly recursive functions such that*
$$(\forall n)(\forall x)(t_n(x) \leq t_{n+1}(x)),$$
then there is a recursive function t such that $\mathcal{C}_t = \bigcup_{n\in\omega} \mathcal{C}_{t_n}$.

Proof. To ensure $\bigcup_{n\in\omega} \mathcal{C}_{t_n} \subseteq \mathcal{C}_t$, we have the requirements
$$P_e \;:\; (\forall_\infty x)(t_e(x) \leq t(x)).$$
To satisfy them, we use as reference the diagonal function
$$d(x) = t_x(x).$$
By the hypothesis on the sequence $\{t_n\}_{n\in\omega}$, d is recursive. If $e \leq x$, then
$$t_e(x) \leq t_x(x) = d(x).$$

Thus, by pushing t sufficiently close to d we can satisfy the first type of requirements. On the other hand, it is not enough to simply define $t = d$, since there could be functions not in $\bigcup_{n\in\omega} \mathcal{C}_{t_n}$, but with some complexity below d almost everywhere.

To ensure $\mathcal{C}_t \subseteq \bigcup_{n\in\omega} \mathcal{C}_{t_n}$, we have the requirements

$$N_e \ : \ (\forall_\infty x)(\Phi_e(x) \leq t(x)) \ \Rightarrow \ \varphi_e \in \bigcup_{n\in\omega} \mathcal{C}_{t_n}.$$

Since their premises refer to the very function we have to build, for the purpose of a construction of t it is better to deal with their (logically equivalent) contrapositives:

$$N_e \ : \ \varphi_e \notin \bigcup_{n\in\omega} \mathcal{C}_{t_n} \ \Rightarrow \ (\exists_\infty x)(t(x) < \Phi_e(x)).$$

These requirements are infinitary, and to satisfy each of them we have to act infinitely often (unlike the requirements considered in Section 2, that could be satisfied by a single action).

Since we always keep t below d, and since if $\varphi_e \notin \bigcup_{n\in\omega} \mathcal{C}_{t_n}$, then

$$(\forall n)(\exists_\infty x)(t_n(x) < \Phi_e(x)),$$

we look at indices n and e such that, for some x,

$$t_n(x) < \Phi_e(x) \leq d(x),$$

and we make an attempt to satisfy N_e by pushing $t(x)$ below $\Phi_e(x)$. However, since a single action is not enough, we cannot simply cancel e after that. What we do is take an infinite sequence $\{t_{\langle e,z\rangle}\}_{z\in\omega}$, look for z and e such that, for some x,

$$t_{\langle e,z\rangle}(x) < \Phi_e(x) \leq d(x),$$

push $t(x)$ below $\Phi_e(x)$, and cancel $\langle e, z\rangle$.

The construction is as follows. At stage x we define $t(x)$. Consider $\langle e, z\rangle$ such that:

- $\langle e, z\rangle \leq x$
- $\langle e, z\rangle$ has not yet been cancelled
- $t_{\langle e,z\rangle}(x) < \Phi_e(x) \leq d(x)$.

If there is such a $\langle e, z\rangle$, choose the least one $\langle e_x, z_x\rangle$, define

$$t(x) = \Phi_{e_x}(x) - 1,$$

and cancel $\langle e_x, z_x \rangle$. If there is no such $\langle e, z \rangle$, let

$$t(x) = d(x).$$

Notice that t is recursive: at stage x we only look for $\langle e, z \rangle \leq x$, the condition $t_{\langle e,z \rangle}(x) < \Phi_e(x) \leq d(x)$ is recursive, and if it holds then $\Phi_e(x)\downarrow$.

It remains to show that each requirement is satisfied.

- *P_e is satisfied, for each e*
  We want to show that $t(x) < t_e(x)$ can happen only finitely many times.

  If $t(x)$ has been defined as $\Phi_{e_x}(x) - 1$, then we had $t_{\langle e_x, z_x \rangle}(x) < \Phi_e(x)$, and so $t_{\langle e_x, z_x \rangle}(x) \leq t(x)$. If $t(x) < t_e(x)$, then $t_{\langle e_x, z_x \rangle}(x) < t_e(x)$ and, by the hypothesis on $\{t_n\}_{n \in \omega}$, $\langle e_x, z_x \rangle < e$. This can happen only finitely often, since each $\langle e_x, z_x \rangle$ is cancelled after stage x.

  If $t(x)$ has been defined as $d(x)$, then from $t(x) < t_e(x)$ we have $x < e$ (since if $e \leq x$ then $t_e(x) \leq d(x)$), and again this can happen only finitely often.

- *N_e is satisfied, for each e*
  Suppose $\varphi_e \notin \bigcup_{n \in \omega} \mathcal{C}_{t_n}$. Then

  $$(\forall z)(\exists_\infty x)(t_{\langle e,z \rangle}(x) < \Phi_e(x)).$$

  There are two cases.

  If $d(x) < \Phi_e(x)$ for infinitely many x, then $(\exists_\infty x)(t(x) < \Phi_e(x))$ because, by definition, $t(x) \leq d(x)$ for every x.

  If $\Phi_e(x) \leq d(x)$ for almost all x, then

  $$(\forall z)(\exists_\infty x)(t_{\langle e,z \rangle}(x) < \Phi_e(x) \leq d(x)).$$

  As soon as $\langle e, z \rangle$ has highest priority (i.e. action has been taken for all smaller indices for which action will ever be taken), we set $t(x) < \Phi_e(x)$ for some $x \geq \langle e, z \rangle$. Since this happens for every z, it follows that $(\exists_\infty x)(t(x) < \Phi_e(x))$. $\square$

Extensions of the Union Theorem are given in Baass and Young [1973], Barashko [1977], Van Emde Boas [1978], and Kummer [1993].

As a typical application, consider the *constant functions* $t_e(x) = e$. Then the Union Theorem tells us that there is a recursive function t such that $\mathcal{C}_t$ consists exactly of the recursive functions with some trivial (bounded) complexity. Such a class can be considered as *a gap at the lower end of the complexity classes w.r.t. any given measure*, and the associated t can be considered as a *minimal growth rate function* (in the sense that it is unbounded and, by construction, every unbounded Φ_e goes above it infinitely often).

Corollary VII.3.8 *Given an r.e. set $\mathcal{C}$ of total recursive functions, we say that*

- *$\mathcal{C}$ is* **self-bounded** *if, for every finite $\mathcal{F} \subseteq \mathcal{C}$, there is $f \in \mathcal{C}$ majorizing every function in $\mathcal{F}$*
- *$\mathcal{C}$ has the* **Ritchie-Cobham property** *if $\mathcal{C} = \bigcup_{t \in \mathcal{C}} \mathcal{C}_t$.*

If $\mathcal{C}$ is self-bounded, then $\bigcup_{t \in \mathcal{C}} \mathcal{C}_t$ is a complexity class. If, moreover, $\mathcal{C}$ has the Ritchie-Cobham property, then $\mathcal{C}$ itself is a complexity class.

Proof. Since the second assertion follows trivially from the first, we let $\mathcal{C} = \{\varphi_{h(e)}\}_{e \in \omega}$ be an r.e. set of total recursive functions, and consider

$$t_n(x) = \max\{\varphi_{h(e)}(x) : e \leq n\}.$$

$\bigcup_{n \in \omega} \mathcal{C}_{t_n}$ is a complexity class by the Union Theorem. It is then enough to show that

$$\bigcup_{n \in \omega} \mathcal{C}_{t_n} = \bigcup_{t \in \mathcal{C}} \mathcal{C}_t.$$

If $t \in \mathcal{C}$, then $t = \varphi_{h(n)}$ for some n, hence t is majorized by t_n, and $\mathcal{C}_t \subseteq \mathcal{C}_{t_n}$. Thus $\bigcup_{t \in \mathcal{C}} \mathcal{C}_t \subseteq \bigcup_{n \in \omega} \mathcal{C}_{t_n}$.

Conversely, since $\mathcal{C}$ is self-bounded, there is $t \in \mathcal{C}$ majorizing every function in $\{\varphi_{h(0)}, \ldots, \varphi_{h(n)}\}$. Hence also t majorizes t_n, and $\mathcal{C}_{t_n} \subseteq \mathcal{C}_t$. Thus $\bigcup_{n \in \omega} \mathcal{C}_{t_n} \subseteq \bigcup_{t \in \mathcal{C}} \mathcal{C}_t$. $\square$

Since self-boundedness for r.e. classes follows from simple closure or growth properties, we have that *for any measure, practically every natural class of recursive functions defines a complexity class*. Typical examples are:

- the *constant functions* $t_e(x) = e$, with diagonal $d(x) = x$
- the *linear functions* $t_e(x) = e \cdot x$, with diagonal $d(x) = x^2$
- the *polynomials* $t_e(x) = x^e$, with diagonal $d(x) = x^x$.

In Chapter VIII we study such complexity classes w.r.t. time and space complexity measures.

The Ritchie-Cobham property for an r.e. class $\mathcal{C}$ is more delicate, and requires proof that each function in $\mathcal{C}$ has a complexity bounded by a function in $\mathcal{C}$, and that each function with a complexity bounded by a function in $\mathcal{C}$ is in $\mathcal{C}$. In particular, the reference to the rate of growth of complexities makes it measure-dependent. For time and space complexity, the first half is usually proved by induction on the definition of $\mathcal{C}$, while the second half follows from simple properties (such as closure under the bounded μ-operator and the existence of a coding and decoding mechanism in the given class). Thus *practically every natural class of recursive functions which is sufficiently rich is a complexity class for measures such as time or space complexity.* Typical examples are the elementary and the primitive recursive functions (see VIII.7.6 and VIII.8.8).

Exercise VII.3.9 *For every measure there is a sequence $\{t_n\}_{n\in\omega}$ of uniformly recursive functions such that*

$$(\forall n)(\forall x)(t_{n+1}(x) \leq t_n(x)),$$

and $\bigcap_{n\in\omega} \mathcal{C}_{t_n}$ is not a complexity class. Thus there is no Intersection Theorem similar to the Union Theorem. (Baass [1972]) (Hint: let g be a monotone recursive function such that

$$(\forall i)(\mathcal{C}_{\Phi_i} \subset \mathcal{C}_{g\circ\Phi_i}),$$

which exists by the Compression Theorem VII.2.8 (see also VII.3.4). Consider the proof of the Speed-Up Theorem VII.2.16, let t_n be the h_n considered there, f be the function built there w.r.t. g, and suppose $\mathcal{C}_t = \bigcap_{n\in\omega} \mathcal{C}_{t_n}$. By construction, $f \in \mathcal{C}_{h_n}$ for all n, and so $f \in \mathcal{C}_t$. Thus, for some index e of f,

$$(\forall_\infty x)(\Phi_e(x) \leq t(x)).$$

By the speed-up property, there is another index i of f such that

$$(\forall_\infty x)[g(x, \Phi_i(x)) \leq \Phi_e(x)].$$

We obtain a contradiction from

$$\mathcal{C}_t \subseteq \mathcal{C}_{h_i} \subseteq \mathcal{C}_{\Phi_i} \subset \mathcal{C}_{g\circ\Phi_i} \subseteq \mathcal{C}_{\Phi_e} \subseteq \mathcal{C}_t,$$

where the second inclusion holds by the proof of VII.2.16.)

Hierarchies for the recursive functions: exhaustiveness ⋆

A trivial modification of the Union Theorem shows that, under natural conditions, any hierarchy of recursive functions indexed by (notations for) recursive ordinals and of length less than ω_1^{ck} cannot be exhaustive (by Rabin's Theorem, since its union is still a complexity class).

Thus, *to obtain a natural exhaustive hierarchy for the recursive functions, we must expect to go all the way through ω_1^{ck}.* To do this, we must choose a system of notation for the recursive ordinals, i.e. a path through $\mathcal{O}$. The following general theorem of Moschovakis (to be proved in Volume III) shows that *there is no natural procedure that can give a hierarchy independent of the path*: if $\{A_a\}_{a\in\mathcal{O}}$ is such that

- $\bigcup_{a\in\mathcal{O}} A_a$ is hyperarithmetical

- the predicate $a \in \mathcal{O} \wedge x \in A_a$ is Π^1_1 (uniformly in x and a),

then there is $b \in \mathcal{O}$ such that $\bigcup_{|a|_\mathcal{O} < |b|_\mathcal{O}} A_a = \bigcup_{a \in \mathcal{O}} A_a$. Thus *a collapse must occur before ω_1^{ck} is reached, and from then on (if not before) uniqueness fails.* In practice, such a collapse usually occurs very early on, e.g. at level ω or ω^2, see Axt [1959], Kreisel [1960a], Feferman [1962], Parikh [1967], Baass and Young [1973].

The problem just discussed is very general, and it affects any natural hierarchy for any hyperarithmetical class of functions. A more specific problem for hierarchies of recursive functions obtained from complexity classes is the following. *If F is any recursive operator such that*

$$(\forall_\infty x)[F(g)(x) \leq F(h)(x)] \ \Rightarrow \ (\forall_\infty x)[g(x) \leq h(x)],$$

then no hierarchy of complexity classes obtained by using F at successor stages and unions at limit stages is exhaustive (Baass and Young [1973]). Precisely, any hierarchy obtained by:

- starting from a given $\mathcal{C}_{t_0}$
- letting $\mathcal{C}_{t_{\alpha+1}} = \mathcal{C}_{F(t_\alpha)}$ at successor stages
- letting $\mathcal{C}_{t_\alpha} = \bigcup_{\beta<\alpha} \mathcal{C}_{t_\beta}$ at limit stages (by using appropriate versions of the Union Theorem)

cannot be exhaustive. To prove this, let f be a function with speed-up w.r.t. F and not in $\mathcal{C}_{t_0}$ (see p. 46 and VII.2.17.a). Then f cannot first appear at limit stages, since we take unions there. It cannot first appear at successor stages either, because if $f \in \mathcal{C}_{t_{\alpha+1}}$, then for some index e of f

$$(\forall_\infty x)[\Phi_e(x) \leq F(t_\alpha)(x)].$$

Since f has speed-up w.r.t. F, it has an index i such that

$$(\forall_\infty x)[F(\Phi_i)(x) \leq \Phi_e(x)],$$

hence

$$(\forall_\infty x)[F(\Phi_i)(x) \leq F(t_\alpha)(x)],$$

and by the hypothesis on F

$$(\forall_\infty x)[\Phi_i(x) \leq t_\alpha(x)],$$

i.e. $f \in \mathcal{C}_{t_\alpha}$. Thus f is not in the hierarchy at all.

For all these reasons, *it is very unlikely that a natural exhaustive hierarchy for all recursive functions can be found* (see Constable [1970], Baass and

Young [1973], and Girard [1987] for more on this topic). In particular, we will have to content ourselves with much less than exhaustive hierarchies, and will instead concentrate (in Chapter VIII) on the more modest goal of finding natural hierarchies for interesting subclasses of the recursive functions. We will also discover a canonical stopping point to the hierarchies developed in such a natural way.

Names for complexity classes $\star$

The Naming Theorem VII.3.6 provided a uniform choice of measured names for all complexity classes. In this subsection we investigate the variety and distribution of general names for such classes.

The next result shows that *a complexity class can have lots of names*, since if $\mathcal{C}_t = \mathcal{C}_{g\circ t}$, then any recursive function between t and $g \circ t$ is a name for $\mathcal{C}_t$.

Theorem VII.3.10 Upward Gap Theorem (Trakhtenbrot [1967], Borodin [1972]) *For any recursive function g such that*

$$(\forall z)(g(x,z) \geq z),$$

there is a recursive function t such that

$$\mathcal{C}_t = \mathcal{C}_{g\circ t}.$$

Proof. Since $(\forall z)(g(x,z) \geq z)$, for every x

$$t(x) \leq g(x, t(x)) = (g \circ t)(x),$$

and thus $\mathcal{C}_t \subseteq \mathcal{C}_{g\circ t}$ is automatically satisfied.

To obtain $\mathcal{C}_{g\circ t} \subseteq \mathcal{C}_t$, we want to rule out any function having some running time between t and $g \circ t$ and ensure, for any e, the global condition

$$(\forall_\infty x)[\Phi_e(x) \leq g(x, t(x))] \Rightarrow (\forall_\infty x)[\Phi_e(x) \leq t(x)].$$

It is actually easier to ensure, for any e, the stronger local (pointwise) condition

$$(\forall_\infty x)[\Phi_e(x) \leq g(x, t(x)) \Rightarrow \Phi_e(x) \leq t(x)],$$

or the logically equivalent one

$$(\forall_\infty x)[\Phi_e(x) \leq t(x) \ \vee \ g(x, t(x)) < \Phi_e(x)].$$

Since, for any given e, each condition has to hold only for almost all x, it is enough to satisfy

$$\Phi_e(x) \leq t(x) \ \vee \ g(x, t(x)) < \Phi_e(x)$$

for all $x \geq e$. This is immediate, by letting

$$t(x) = \mu z\,((\forall e \leq x)[\Phi_e(x) \leq z \ \vee \ g(x,z) < \Phi_e(x)]).$$

The function t is total because to define $t(x)$ we only consider $e \leq x$, and thus there are only finitely many convergent values of $\Phi_e(x)$ to avoid. Thus any z larger than all of them satisfies the condition of the definition, and hence there is a smallest such z.

The function t is recursive because, for each e and z, the condition

$$\Phi_e(x) \leq z \ \vee \ g(x,z) < \Phi_e(x)$$

is recursive, and we only have to test it for $e \leq x$.

It remains to prove $\mathcal{C}_{g \circ t} \subseteq \mathcal{C}_t$. If $f \in \mathcal{C}_{g \circ t}$, then there is an index e of f such that

$$(\forall_\infty x)[\Phi_e(x) \leq g(x, t(x))].$$

By definition of t,

$$(\forall_\infty x)[\Phi_e(x) \leq t(x)]$$

and hence $f \in \mathcal{C}_t$. $\square$

The Upward Gap Theorem has a number of interesting consequences. First, its proof actually shows that no Φ_e can be between t and $g \circ t$ for almost every x, and hence there are arbitrarily large gaps in which no step-counting function falls. Thus *the step-counting functions are very sparse among the recursive functions*. Another manifestation of this fact is that the step-counting functions are (by definition) a measured set, hence small in a topological sense (see VII.2.10.c).

Second, the Upward Gap Theorem provides another proof of the fact (discussed on p. 43) that *no computer can be optimal for every purpose*. For example, let M_g and M_b be two Turing machines computing the universal function $\varphi(e,x) \simeq \varphi_e(x)$, of which the former is very good while the latter is hopelessly bad. Let $\{\Phi_e\}_{e\in\omega}$ and $\{\Psi_e\}_{e\in\omega}$ be the time complexity measures relative to M_g and M_b, respectively. By recursive relatedness (VII.1.15), there is a recursive function g such that, for every e and almost every x, if $\varphi_e(x)\downarrow$, then

$$\Psi_e(x) \leq g(x, \Phi_e(x)).$$

We can always suppose that g is monotone and such that $g(x,z) \geq z$ (otherwise it is enough to consider $z + \max_{y \leq z} g(x,y)$ in place of $g(x,z)$). By the Upward Gap Theorem relative to g, there is a recursive function t such that

$$(\forall_\infty x)[\Psi_e(x) \leq g(x,t(x))] \ \Rightarrow \ (\forall_\infty x)[\Psi_e(x) \leq t(x)], \qquad \text{(VII.4)}$$

i.e. $\mathcal{C}_t^\Psi = \mathcal{C}_{g \circ t}^\Psi$. If, for some e,

$$(\forall_\infty x)(\Phi_e(x) \leq t(x)),$$

then, by recursive relatedness and monotonicity of g,

$$(\forall_\infty x)[\Psi_e(x) \leq g(x, \Phi_e(x)) \leq g(x, t(x))].$$

By VII.4,

$$(\forall_\infty x)(\Psi_e(x) \leq t(x)),$$

i.e. $\mathcal{C}_t^\Phi = \mathcal{C}_t^\Psi$. In other words, whatever the good computer M_g computes within the bound t, so does the bad computer M_b!

Exercise VII.3.11 *The function t of the Upward Gap Theorem can be arbitrarily large.* (Hint: given h recursive, add the clause $h(x) \leq z$ to the definition of t.)

Constable [1972] shows that the theorem holds not only for recursive functions g, but also for total recursive operators G such that

$$(\forall z)[G(f)(z) \geq f(z)].$$

In this case there is a recursive function t such that

$$\mathcal{C}_t = \mathcal{C}_{G(t)}.$$

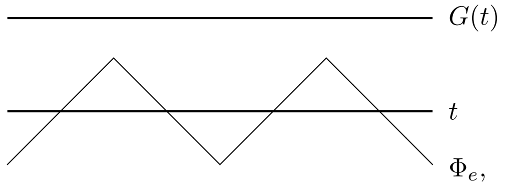
The proof of this result does not have the simple look of that of VII.3.10 because the stronger local result proved there, namely that there is a recursive function t such that, for all e,

$$(\forall_\infty x)[\Phi_e(x) \leq g(x, t(x)) \ \Rightarrow\ \Phi_e(x) \leq t(x)]$$

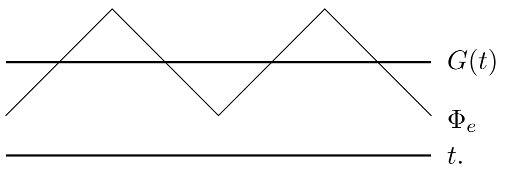
fails for total recursive operators (Blum [1967]). What is possible to prove in the general case is the weaker global result that there is a recursive function t such that, for all e,

$$(\forall_\infty x)[\Phi_e(x) \leq G(t)(x)] \ \Rightarrow\ (\forall_\infty x)[\Phi_e(x) \leq t(x)].$$

In other words, the following picture is ruled out:



but not the following:



See Young [1973], Di Paola [1978], Yang [1979] and Calude [1988] for proofs and extensions of the Upward Gap Theorem.

We turn now to a dual version of the Upward Gap Theorem, and for the rest of this section we use the following convention:

Definition VII.3.12 $g^{-1}(x) = \mu z\,(g(z) \geq x)$.

We do not simply let

$$g^{-1}(x) = \mu z\,(g(z) = x)$$

since, otherwise, g^{-1} would be total only when the range of g is ω. With the previous definition, g^{-1} is instead total whenever the range of g is unbounded.

Notice that, for every x,

$$x \leq g(g^{-1}(x)).$$

If $g(z) \geq z$ (in which case the range of g is unbounded and g^{-1} is thus total) then we also have

$$g^{-1}(x) \leq x.$$

A literal analogue of the Upward Gap Theorem for inverse functions would only require *some* name t of some class $\mathcal{C}$, such that $\mathcal{C}_{g^{-1}\circ t} = \mathcal{C}_t$. The next result is much stronger, and holds for *every* name of some class $\mathcal{C}$.

Theorem VII.3.13 Downward Gap Theorem (Borodin [1972], Chow [1972], Constable [1973]) *For any monotone recursive function g such that*

$$(\forall z)(g(z) \geq z),$$

there is a complexity class $\mathcal{C}$ such that, for any recursive function t,

$$\mathcal{C} = \mathcal{C}_t \;\Rightarrow\; \mathcal{C}_{g^{-1}\circ t} = \mathcal{C}_t.$$

Proof. Since $(\forall z)(g(z) \geq z)$, for every x

$$(g^{-1} \circ t)(x) = g^{-1}(t(x)) \leq t(x),$$

and thus $\mathcal{C}_{g^{-1} \circ t} \subseteq \mathcal{C}_t$ is automatically satisfied.

If $\mathcal{C}_{g^{-1} \circ t} \subset \mathcal{C}_t$, there is an index e such that

1. $(\forall_\infty x)[\Phi_e(x) \leq t(x)]$
2. $(\exists_\infty x)[g^{-1}(t(x)) < \Phi_e(x)]$.

From 2, by properties of g^{-1} and monotonicity of g,

$$(\exists_\infty x)[t(x) \leq g(g^{-1}(t(x))) \leq g(\Phi_e(x))].$$

If we define (e.g. by Rabin's Theorem, done uniformly) a function $\varphi_{f(e)}$ such that all its complexities are above $g \circ \Phi_e$ almost everywhere, then they are above $t(x)$ infinitely often, and $\varphi_{f(e)} \notin \mathcal{C}_t$.

From 1, $\mathcal{C}_{\Phi_e} \subseteq \mathcal{C}_t$. If we find a recursive function h such that $\varphi_{f(e)} \in \mathcal{C}_{h \circ \Phi_e}$, and a complexity class $\mathcal{C}$ closed under h in the sense that

$$\mathcal{C}_{\Phi_e} \subseteq \mathcal{C} \Rightarrow \mathcal{C}_{h \circ \Phi_e} \subseteq \mathcal{C},$$

then we have that $\mathcal{C} \neq \mathcal{C}_t$. Otherwise $\mathcal{C}_{\Phi_e} \subseteq \mathcal{C}_t = \mathcal{C}$ and hence $\mathcal{C}_{h \circ \Phi_e} \subseteq \mathcal{C}_t$, while $\varphi_{f(e)} \in \mathcal{C}_{h \circ \Phi_e}$ but $\varphi_{f(e)} \notin \mathcal{C}_t$.

If we define f, h and $\mathcal{C}$ as above we thus have, automatically, that

$$\mathcal{C}_{g^{-1} \circ t} \neq \mathcal{C}_t \Rightarrow \mathcal{C} \neq \mathcal{C}_t,$$

and hence we have proved (the contrapositive of the statement of) the theorem.

As in the proof of the Compression Theorem (see VII.2.10.a), one can obtain recursive functions f and h such that:

1. $\varphi_{f(e)}$ is total if and only if Φ_e is total
2. $(\forall i)[\varphi_{f(e)} \simeq \varphi_i \Rightarrow (\forall_\infty x)(g(\Phi_e(x)) < \Phi_i(x))]$
3. $(\forall_\infty x)[\Phi_{f(e)}(x) \leq h(x, \Phi_e(x))]$.

In particular, whenever Φ_e is total then $\varphi_{f(e)}$ is total by 1, is not in $\mathcal{C}_t$ by 2, and is in $\mathcal{C}_{h \circ \Phi_e}$ by 3.

It only remains to find $\mathcal{C}$ closed under h in the sense above. We let $\mathcal{C} = \mathcal{C}_{t^*}$, and define t^* such that

$$(\forall_\infty x)[\Phi_e(x) \leq t^*(x)] \Rightarrow (\forall_\infty x)[h(x, \Phi_e(x)) \leq t^*(x)].$$

As in the proof of the Upward Gap Theorem, we can actually ensure

$$(\forall_\infty x)[\Phi_e(x) \leq t^*(x) \;\Rightarrow\; h(x, \Phi_e(x)) \leq t^*(x)]$$

by defining

$$t^*(x) = \mu z\,((\forall e \leq x)[z < \Phi_e(x) \;\vee\; h(x, \Phi_e(x)) \leq z]). \quad \Box$$

The strong formulation of VII.3.13 shows that *the phenomenon of downward gaps is real and name-independent*: given g, there are classes that g^{-1} applied to any of their names cannot shrink. On the other hand, *the phenomenon of upward gaps is apparent and name-dependent*: given g, there are classes that g cannot extend if applied to some of their names (VII.3.10); but there is a fixed g that would extend any class, if applied to an appropriate name (by VII.3.4 and VII.3.6).

Exercise VII.3.14 *The class $\mathcal{C}$ of the Downward Gap Theorem can have an arbitrarily large name.* (Hint: given h recursive, add the clause $h(x) \leq z$ to the definition of t^*.)

The phenomenon of gaps can be visualized as follows. A complexity class is a class of recursive functions that can be computed in some bounded complexity. Since complexities are measured by (special) step-counting functions Φ_e, and bounds by (arbitrary) recursive functions t, there is a certain freedom in the choice of a name t for a class $\mathcal{C}$. In particular, functions that do not differ significantly (i.e. infinitely often) from the point of view of step-counting functions, define the same complexity class. If we picture the set of names of a given class as a strip of the plane, with the intuition that any recursive function whose graph lies in that strip is a name for the given class, then the results of this section give us the following picture:

- by the Upward Gap Theorem VII.3.10, the height of the strip can be arbitrarily large
- by the Naming Theorem VII.3.6, the strip always contains a measured name
- by the existence of jump operators for measured sets VII.3.4 (applied to the measured set given by the Naming Theorem), there is a fixed recursive factor g such that a measured name must lie around the top of the strip, by at most the factor g
- by the Downward Gap Theorem VII.3.13, there may be large gaps in which no measured name lies.

VII.4 Time and Space Measures

In the previous sections we have developed a general theory of complexity, valid for every measure satisfying the minimal conditions of Definition VII.1.13. We now concentrate on special properties of the two natural measures of time and space complexity. We will see that, in general, *space complexity is better behaved than the more refractory time complexity* (recall, in particular, that space complexity classes are r.e. and closed under intersection, see VII.3.2.d and VII.3.3.d).

One-tape Turing machines

In Section I.4 we have discussed the stability of the notion of a Turing machine w.r.t. the class of associated computable functions. In particular, we have noticed how the same class (namely, the class of recursive functions) is obtained, independently of restrictions on the 'hardware' (number of states, tapes, and heads), the 'software' (number of symbols of the alphabet), and the (deterministic or nondeterministic) nature of the computations.

The notion of Turing machine computability is however independent of the particular model of Turing machine only globally. Different models provide different local computations, and thus it is conceivable (and actually provable) that the associated measures may also differ. We then fix a particular model, to which we refer in this section and in the next chapter. Our choice is quite restrictive, and we discuss below the relationships with other models.

Our model is simply the **one-tape (on-line) Turing machine** defined in Section I.4, and the definition of time complexity classes is a straightforward conceptualization of the length of a computation.

Definition VII.4.1 Time Complexity Classes (Yamada [1962], Rabin [1963], Hartmanis and Stearns [1965]) *We denote by* $\boldsymbol{\mathcal{C}_t^T}$ *or* **TIMEF** $[\,\boldsymbol{t}\,]$ *the class of* **functions computable in time** $\boldsymbol{t}$*, i.e. the class of all recursive functions computable by some Turing machine with a single tape infinite in both directions (used for input, output and working activity), and not doing more than* $t(x)$ *moves during a computation on input* x.

We denote by **TIME** $[\,\boldsymbol{t}\,]$ *the restriction of* TIMEF $[\,t\,]$ *to 0,1-valued functions, i.e. to sets.*

The definition of space complexity classes is slightly complicated by the fact that the use of the same model as above would prevent the consideration of small space measures, making the consideration of bounds smaller than the representation of the inputs or the outputs impossible. Since space measure is intended to be a conceptualization of the amount of memory needed for side work, we modify the previous model and consider a **one-tape off-line Turing machine**, in which inputs and outputs are isolated from the computation and confined to separate tapes.

Definition VII.4.2 Space Complexity Classes (Trakhtenbrot [1956], Myhill [1960], Ritchie [1963], Hartmanis, Lewis and Stearns [1965]) *We denote by* $\boldsymbol{\mathcal{C}^S_s}$ *or* **SPACEF [*s*]** *the class of* **functions computable in space *s***, *i.e. the class of all recursive functions computable by some Turing machine with:*

- *a read-only input tape*
- *a working tape infinite in both directions*
- *a write-only output tape*

and not scanning more than $s(x)$ *cells of the working tape during a computation on input* x.

We denote by **SPACE [*s*]** *the restriction of* SPACEF [s] *to 0,1-valued functions, i.e. to sets.*

We have seen that, in general, certain pathologies (for example, gaps of any prescribed width) can occur if we consider unrestricted measure bounds. Some of the results to be proved in this section (for example, hierarchy theorems) can thus hold only under suitable restrictions. The first one that comes to mind is to restrict attention to the step-counting functions. Since it turns out that this restriction is indeed sufficient for many results of this section, special names have been introduced to refer to these functions.

Definition VII.4.3 *A function* f *is called* **time constructible** *if it is a step-counting function for the time complexity measure, i.e. if* $f \simeq T_e$ *for some* e.

Similarly, a function f *is called* **space constructible** *if it is a step-counting function for the space complexity measure, i.e. if* $f \simeq S_e$ *for some* e.

Exercises VII.4.4 a) *The space and time constructible functions are closed under composition, sum, product and exponential.*

b) *The functions* $(\log|x|)^n$, $|x|^n$, $2^{n\cdot|x|}$ *and* $2^{|x|^n}$ *are space and time constructible.*

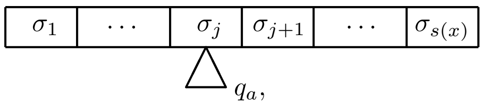
c) *A function* s *is space constructible if and only if it is computable in space* s. (Hint: if any function is computable exactly in space s, then s is space constructible. And if s is computable in space s, then it is also computable in space exactly s, by using the output to mark down the needed space. Conversely, if s is space constructible, there is a function f computable exactly in space s. By discharging the output of f and writing down instead the number of cells used during the computation of f, one sees that s in computable in space s.)

Kobayashi [1985] has proved the analogue of part c) for time, for any sufficiently large function (the notion of largeness depends on the Turing machines model as in the hypothesis of the Time Linear Speed-Up Theorem, see VII.4.9 and VII.4.11).

A special restriction for space bounds comes from the hypothesis of the following computation, which is used in many results of this section.

Proposition VII.4.5 *If $s(x) \geq \log |x|$, then the number of possible configurations on input x of a Turing machine working in space $s(x)$ is bounded by $c^{s(x)}$, for some constant c.*

Proof. The main computation of the proof is an evaluation of the possible configurations of the working tape



which can be represented by sequences

$$\sigma_1 \cdots \sigma_j q_a \sigma_{j+1} \cdots \sigma_{s(x)}$$

in which all consecutive symbols σ_i written in the relevant portion of the tape are indicated, together with the state and head position (the latter shown by the position of the state symbol among the alphabet symbols). Notice that we only consider sequences of $s(x)$ symbols because M works in space s, and thus the relevant portion of the working tape is bounded by $s(x)$.

Suppose M has an alphabet of n symbols and m states. Then the possible sequences $\sigma_1 \cdots \sigma_{s(x)}$ are $n^{s(x)}$, the possible head positions are $s(x)$, and the possible configurations of the working tape are thus $m \cdot s(x) \cdot n^{s(x)}$.

Each of these configurations could in principle occur for any input. Since the input tape is nonerasable, its configurations can be represented simply by state and head positions, and hence they are bounded by $m \cdot |x|$.

Thus the total number of possible configurations of M is bounded by

$$m \cdot |x| \cdot s(x) \cdot n^{s(x)}$$

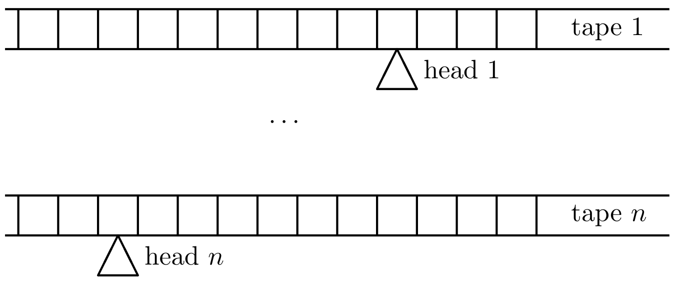
(notice that m is used only once, because the states of the input and working tapes must be the same). But $s(x) \leq 2^{s(x)}$ in general, and $|x| \leq 2^{s(x)}$ by the hypothesis on s, and thus there is a constant c such that

$$m \cdot |x| \cdot s(x) \cdot n^{s(x)} \leq c^{s(x)}. \quad \Box$$

Other Turing machine models ⋆

We briefly discuss here the relationships between the one-tape Turing machine model considered above and other models. First we specify, at least informally, the other models that we refer to.

An ***n*-tape Turing machine** (for $n \geq 1$) can be pictured as follows:



and it consists, as in definition I.4.1, of:

- a finite alphabet $\{s_0, \ldots\}$
- a finite set of states $\{q_0, \ldots\}$
- a finite consistent set of instructions $\{I_0, \ldots\}$.

For one-tape Turing machines, an instruction

$$q_a s_b s_c q_d j$$

tells the machine in state q_a and reading s_b to print s_c in place of s_b, change its state to q_d and (depending on whether $j = 0, 1, 2$) stay still, move one cell to the right, or move one cell to the left.

For n-tape Turing machines, an instruction

$$q_a \left\{ \begin{array}{c} s_{b_1} \\ \cdots \\ s_{b_n} \end{array} \right\} \left\{ \begin{array}{c} s_{c_1} \\ \cdots \\ s_{c_n} \end{array} \right\} q_d \left\{ \begin{array}{c} j_1 \\ \cdots \\ j_n \end{array} \right\}$$

tells the machine in state q_a and reading s_{b_i} on the i-th tape (for $1 \leq i \leq n$) to print s_{c_i} in place of s_{b_i}, change its state to q_d and (depending on whether $j_i = 0, 1, 2$) leave the head scanning the i-th tape still, move it one cell to the right, or move it one cell to the left.

The notion of **computability by an *n*-tape Turing machine** is defined by a straightforward generalization of I.4.2, the details of the definition being arbitrary (in the sense that the class of functions thus defined is widely independent of them, as long as the machine carries configurations coding the inputs in some fixed way, to configurations coding the output in some fixed way). For definiteness, we require the heads on the n tapes to all be aligned at the beginning of the computation, with the head on the first tape scanning the first blank cell to the right of the input.

Time complexity associated with computability by an n-tape Turing machine is the number of moves (in the sense of applications of instructions) performed during the computation, while **space complexity** is the maximum number of cells used on any single working tape during the computation.

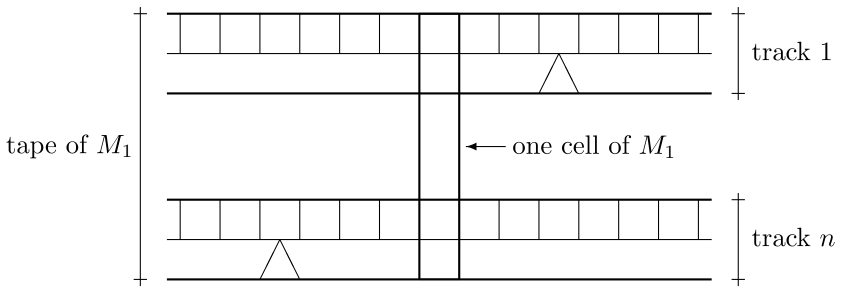
The notion of **computability by a multitape Turing machine** is defined as computability by some n-tape Turing machine, for some $n \geq 1$. In other words, no fixed bound is imposed on the number of tapes.

We are now ready for our first simulation result, which is a special case of the stability of the notion of Turing machine computability.

Proposition VII.4.6 Simulation of Multitape Turing Machines (Hartmanis and Stearns [1965], Hartmanis, Lewis and Stearns [1965])

1. *A function computable in space s by a multitape Turing machine is computable in space s in the sense of VII.4.2.*
2. *A function computable in time t by a multitape Turing machine is computable in time $c \cdot t^2$ in the sense of VII.4.1, for a fixed constant c independent of t.*

Proof. Consider a Turing machine M with n working tapes ($n > 1$). We build a one-tape Turing machine M_1 equivalent to M, in a natural way. The unique tape of M_1 will be seen as consisting of n double tracks: the upper part of each track will reproduce one tape of M, while the lower part will indicate the position of the corresponding head of M:



This can be achieved by letting each symbol of the alphabet of M_1 code a sequence of $2n$ pieces of information, in such a way that from a symbol in the cell scanned by M_1 one can, for each tape of M, read off the symbol in the corresponding cell and know whether this is the cell scanned by the corresponding head.

To simulate one move of M, we need to know the content of all its n scanned cells (one for each tape). Thus M_1 has to make a sweep in one direction, and to record the content of each scanned cell. For example, we can always suppose that at the beginning of the simulation of one move of M, the head of M_1 is on the cell containing the rightmost head of M, and that the state of M_1 successively records how many heads of M have already been considered, so that M_1 knows when the last head of M has been hit (since their number n is fixed).

When all scanned cells of M have been considered, M_1 has the necessary information to simulate one move of M. But one move of M may change the content of any of its scanned cells, as well as the positions of the corresponding heads. Thus M_1 needs to make another sweep in the opposite direction, to record these changes.

Since this procedure does not take more space than M, part 1 is proved. But the simulation of one move of M requires many moves: M_1 makes one sweep from the rightmost head of M to the leftmost head, and then it comes back. On the way back, each time it hits a head of M it might need to do two more moves for each track, going one cell to the left or to the right (to record the fact that the head of M under consideration has moved), and then back to the previous position (since there might be more than one head of M on the same position of the corresponding tapes, and hence on different tracks of the same cell of M_1).

Since two heads of M could always go in opposite directions, they can be $2x$ cells apart after x moves of M. Thus M_1 may need $4x + 2n$ moves to simulate the x-th move of M_1. To simulate the first x moves, M_1 then needs at most

$$(\sum_{i \leq x} 4i) + 2xn = 2x^2 + 2x(1+n)$$

moves. Let c be a constant such that, for every x,

$$2x^2 + 2x(1+n) \leq c \cdot x^2.$$

If M computes a function in time t, then M_1 computes the same function in time $c \cdot t^2$. □

If $\lim_{x\to\infty} \dfrac{t(x)}{\max\{|x|, |f(x)|\}} = \infty$, then the reference to the constant c can be eliminated since, by VII.4.11, M can be supposed to compute f in time

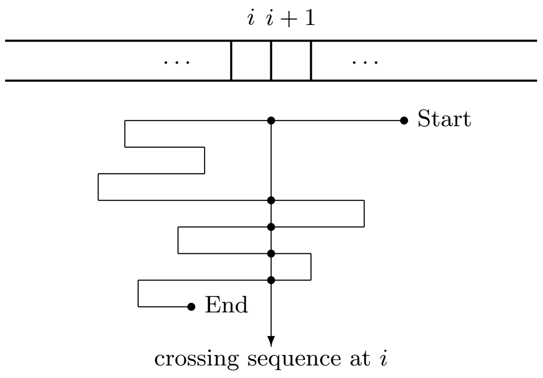
$\frac{1}{\sqrt{c}} \cdot t$, and thus M_1 computes f in time

$$c \cdot \left(\frac{1}{\sqrt{c}} \cdot t \right)^2 = t^2 .$$

Exercises VII.4.7 Two-tape Turing machines.

a) *VII.4.6 is the best possible, since there are 0,1-valued functions computable in time proportional to $|x|$ by a two-tape Turing machine, but not computable in time less than $|x|^2$ by any one-tape Turing machine.* (Rabin [1963], Bardzin [1965], Hennie [1965])[2] (Hint: consider the set A of **palindromes**, i.e. those numbers that have symmetric binary representations, and let f be the characteristic function of A. Then f is computable in time proportional to $|x|$ by a two-tape Turing machine, by just copying the input on the second tape, moving the heads to opposite extremes, and comparing the digits in opposite order.

To prove that f is not computable in time less than $|x|^2$ by any one-tape Turing machine, we define the **crossing sequence at i** as the sequence of states in which the Turing machine finds itself when going across the boundary between cells i and $i + 1$, plus the states corresponding to moves in which the head does not change its position, possibly repeated if they do not change. For definiteness, we suppose that the Turing machine enters the new state before moving the head. We also require that the machine stops at the right of the cells where the input was written (this only increases the time needed for the computation by a constant factor, e.g. at most by doubling it).



Note that if a one-tape Turing machine computes an output on a certain input $w_1 * w_2$, then it computes the same output on the input $w_3 * w_2$, whenever $|w_1| = |w_3|$ and the crossing sequences at the end of w_1 and w_3 are the same.

Let M be a Turing machine with binary input alphabet and s states, that computes f. Among the strings of 0's and 1's of length $2x$, 2^x are in A: one half of the

[2]The result holds for the one-tape on-line Turing machine model we are using for time complexity. Maass [1985b], and Li and Vitányi [1988] have proved the same result also for the one-tape off-line Turing machine model we are using for space complexity.

string is arbitrary and of length x, the other half must be symmetric. For $1 \leq i \leq x$, let

$l(i) =$ average length of the crossing sequences at i, for words in A of length $2x$.

Then the number of distinct crossing sequences of length n is $\leq s^n$, and the number of distinct crossing sequences of length $\leq 2l(i)$ is

$$\leq \sum_{n \leq 2l(i)} s^n \leq s^{2l(i)+1}.$$

At least half of the strings of length $2x$ in A, i.e. at least 2^{x-1} of them, must have crossing sequences of length not more than twice the average, thus at least $\frac{2^{x-1}}{s^{2l(i)+1}}$ strings of length $2x$ in A have the same crossing sequence at i. The possible strings filling up the cells with positions from $i+1$ to x are 2^{x-i}, and if

$$\frac{2^{x-1}}{s^{2l(i)+1}} > 2^{x-i},$$

then there are two strings differing in the first i places, and with the same crossing sequence at i. But this is impossible: if $w_1 * w_2$ is in A, $|w_3| = |w_1|$, and the crossing sequences of $w_1 * w_2$ and $w_3 * w_2$ are the same at the end of w_1 and w_3, then $w_3 * w_2$ is also in A, but $w_3 * w_2$ must be symmetric, and thus $w_1 = w_3$.

This proves that

$$\frac{2^{x-1}}{s^{2l(i)+1}} \leq 2^{x-i},$$

i.e.

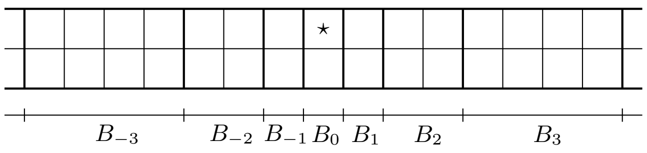
$$l(i) \geq \frac{i-1}{2 \cdot \log_2 s} - \frac{1}{2}.$$

Notice that the time needed to compute f on any input is at least the sum of the lengths of the crossing sequences at each position on the input. And $\sum_{i \leq x} l(i)$, which bounds a polynomial of degree 2 in x, is the sum of the average lengths of crossing sequences, hence a lower bound of the time on some input of length $2x$.)

b) *A function computable in time $t(x)$ by a multitape Turing machine is computable in time $c \cdot t(x) \cdot \log t(x)$ by a two-tape Turing machine, for a fixed constant c independent of t.* (Hennie and Stearns [1966]) (Hint: given an n-tape Turing machine M computing $f(x)$ in time $t(x)$, we build an equivalent two-tape Turing machine M_1. The auxiliary tape of M_1 is kept for scratch work, while the principal one has n double-tracks, coding the content of the n tapes of M. Each track is divided into blocks

$$\cdots, B_{-2}, B_{-1}, B_0, B_1, B_2, \cdots$$

such that B_i and B_{-i} have length 2^{i-1}:



We show how to code one tape of M by the corresponding double track of the principal tape of M_1. B_0 always contains the symbol scanned by the given head of M on the lower part, and a special mark on the upper one. Thus the full cell scanned by M_1 gives the content of all cells scanned by the various heads of M, and M_1 can simulate one move of M by just looking at one cell. At the beginning, the lower part of the track is a copy of the corresponding tape of M, while the upper part is empty (except for the special mark). When the head of M moves in one direction, we shift part of the content of the track of M_1 in the other direction, by moving information in the upper part if needed.

By a *B_i-operation* we mean the following set of moves, intended to shift information to the right in a more efficient way than simply shifting everything one cell to the right. Search for the first block B_i whose upper part is empty and move the information stored in the preceding block B_{i-1} into B_i (on the upper part if the bottom is full, on the bottom part otherwise). Then move the information stored in B_{i-2} into the bottom part of B_{i-1}, and so on. To perform this, we only need a time proportional to the length 2^{i-1} of B_i, since we have another tape for scratch work.

Then also move the upper part of B_{-i} if not empty, and the lower part otherwise, to the right, by making the necessary shifts in all blocks between B_{-i} and B_{-1}. This part can also be performed in time proportional to the length 2^{i-1} of B_i, by using the auxiliary tape both to measure the distance between B_0 and B_i, to find B_{-i}, and then to store the information to be moved, as before.

A *B_{-i}-operation* is defined in a similar way, with the intent of moving information in the other direction. The cost of one B_i- or B_{-i}-operation is thus of $c \cdot 2^i$ moves of M_1, for some c.

To perform two consecutive B_i-operations M needs to move at least 2^{i-1} times in between, since it takes this much for $B_1, \ldots, B_{i-1}$ to fill their upper parts, which are empty after one B_i-operation. Thus, if M makes $t(x)$ moves, then M_1 performs not more than $\frac{t(x)}{2^{i-1}}$ B_i- or B_{-i}-operations.

To make it possible for M_1 to perform one B_i- or B_{-i}- operation for the first time, M needs to make at least 2^{i-1} moves. Thus, if M makes $t(x)$ moves, then M_1 performs B_i- or B_{-i}-operations only for $i \leq 1 + \log t(x)$.

Then M_1 makes at most

$$\sum_{i \leq 1+\log t(x)} c \cdot 2^i \cdot \frac{t(x)}{2^{i-1}} \leq 4c \cdot t(x) \cdot \log t(x)$$

moves.)

The simulation results we have proved are barely the tip of an iceberg, and much work has been done on the trade-off of the size of the alphabet, number of tapes, number of heads per tape, number of dimensions of tapes, and possible directions of head movement. In another direction, variations of the basic Turing machine model have been considered, by adding pushdown stores, counters, various kinds of stacks, recursive calls, and parallel calls. Finally, simulations by Turing machines of other models of computations, most notably various kinds of random access machines (see p. I.64) using different primitive assignments such as Boolean and arithmetical operations, have also been considered.

For a survey of these topics and references, see Van Emde Boas [1990]. For treatments, see Aho, Hopcroft and Ullman [1974], Hopcroft and Ullman [1979], Balcázar, Díaz and Gabarró [1990], Wagner and Wechsung [1986] and Papadimitriou [1994].

Linear Speed-Up

We have noticed on p. 37 that a total recursive function always has a best complexity modulo some recursive function g. We now prove that, for the time and space complexities, g must be more than linear. In other words, every (sufficiently large) recursive function is always speedable by a constant factor.

The proofs of the next two results amount to the simple observation that space and time can be economized by enlarging the alphabet and the number of states of a given Turing machine.

Theorem VII.4.8 Space Linear Speed-Up Theorem (Hartmanis, Lewis and Stearns [1965]) *If $c > 0$ is a real constant, then*

$$\text{SPACEF}\,[\,s\,] = \text{SPACEF}\,[\,c \cdot s\,].$$

Proof. It is enough to restrict attention to c such that $0 < c \leq 1$, since if $c > 1$ one then obtains the result by considering $\frac{1}{c}$ and $c \cdot s$ in place of c and s.

If $c \leq 1$, then $c \cdot s(x) \leq s(x)$ for all x and $\text{SPACEF}\,[\,c \cdot s\,] \subseteq \text{SPACEF}\,[\,s\,]$.

If $c > 0$, we prove $\text{SPACEF}\,[\,s\,] \subseteq \text{SPACEF}\,[\,c \cdot s\,]$. Let M be a Turing machine computing f in space s. By enlarging the alphabet we can easily build a new Turing machine M_1 which simulates M and such that each symbol of the working tape of M_1 codes n adjacent symbols of the working tape of M, for any given n. Then M_1 computes $f(x)$ in $\frac{s(x)}{n}$ space. Since we want

$$\frac{s(x)}{n} \leq c \cdot s(x),$$

we only have to choose n such that $\frac{1}{n} \leq c$, which is possible because $c > 0$.

Note that if $s(x) < n$, then M_1 computes $f(x)$ by using only one cell. We thus have to state by convention that

$$(c \cdot s)(x) = \max\{1, c \cdot s(x)\}. \quad \square$$

The Linear Speed-Up Theorem for Space should not be taken literally as saying that the amount of space used in a computation can always be (linearly) compressed by using a different machine. Rather, it should be seen as showing that, even in the abstract context of Turing machines, a notion of space measure that accounts only for the number of cells used in a computation (and not for the size of the alphabet) is quite unrealistic. After all, calling 'n symbols of a given alphabet' a 'single symbol of a new alphabet' is simply a rhetorical device: when one reaches the technological limits of symbol recognition, reading a symbol of the compressed alphabet would still require n atomic acts.

The analogue of VII.4.8 for Time cannot be as neatly stated in terms of complexity classes, since it depends on how the time bound relates to the lengths of the input and of the output.

Theorem VII.4.9 Time Linear Speed-Up Theorem (Hartmanis and Stearns [1965]) *If $c > 0$ is a real constant, then*

$$\lim_{x\to\infty} \frac{t(x)}{\max\{|x|^2, |f(x)|^2\}} = \infty \ \wedge \ f \in \mathrm{TIMEF}\,[\,t\,] \ \Rightarrow \ f \in \mathrm{TIMEF}\,[\,c \cdot t\,].$$

Proof. As in the previous proof, it is enough to show that if $f \in \mathrm{TIMEF}\,[\,t\,]$, then $f \in \mathrm{TIMEF}\,[\,c \cdot t\,]$ for $0 < c \leq 1$. Let M be a Turing machine computing f in time t. As above, the idea is to build a new Turing machine M_1 which simulates M and does in a single move what M does in n moves, for any given n. Then M_1 computes $f(x)$ in $\frac{t(x)}{n}$ time. The details of the implementation are, however, a little more complicated than before, and it actually turns out that we simulate n moves of M in 8 moves of M_1, instead of just one.

Indeed, to simulate n moves of M we cannot just look at the cell scanned by M: at each move of M, the head can move left or right, and thus in general we need to look at $2n + 1$ cells (the presently scanned one, plus n to the left and n to the right) to be able to predict the behavior after n moves.

The simplest coding of tape is probably that already used in the proof of VII.4.8, where each cell or symbol of M_1 codes n adjacent cells or symbols of M. The state of M_1 can record which cell of M in a given cell of M_1 is actually scanned, and thus this takes no additional moves.

It is clear that $2n + 1$ cells of M are contained in at most 3 cells of M_1, namely the one currently scanned, and the two adjacent ones. So *4 moves* (e.g. left, right, right, left) are enough for M_1 to scan all the cells containing information needed to simulate n moves of M. With this information available, M_1 can thus proceed to the simulation. Again, *4 moves* are needed in general to print the new tape of M_1 (after n moves of M), since the portion of the tape of M that can change is the same portion of $2n+1$ cells, and hence the portion of the tape of M_1 involved is the same portion of 3 cells. Thus 8 moves of M_1 can simulate n moves of M.

It remains to determine the value of n that would give the result, i.e. such that M_1 will compute $f(x)$ in time $c \cdot t(x)$.

- Given the input x on M, we have to translate it into an input for M_1 carrying the same information. If x requires $|x|$ space on M, then it will require $\frac{|x|}{n}$ space on M_1. Since we are working with single-tape Turing machines, to translate the input x (while at the same time erasing the original one) we have to go back and forth through its representation, and this requires $|x|^2 + \left(\frac{|x|}{n}\right)^2$ moves.
- Given the output $f(x)$ on M_1, we have to translate it back into an output for M. If $f(x)$ requires $|f(x)|$ space on M, then it requires space $\frac{|f(x)|}{n}$ space on M_1, and thus the translation requires now $|f(x)|^2 + \left(\frac{|f(x)|}{n}\right)^2$ moves.

Thus M_1 will simulate $t(x)$ moves of M on input x in

$$|x|^2 + \frac{|x|^2}{n^2} + 8 \cdot \frac{t(x)}{n} + |f(x)|^2 + \frac{|f(x)|^2}{n^2}$$

moves. Since we want this to be at most $c \cdot t(x)$, we need to majorize the terms in $|x|$ and $|f(x)|$ by multiples of $t(x)$. By the hypothesis

$$\lim_{x \to \infty} \frac{t(x)}{\max\{|x|^2, |f(x)|^2\}} = \infty,$$

for almost every x

$$\frac{t(x)}{|x|^2} \geq \frac{3}{c} \geq 1$$

(notice that $\frac{3}{c} \geq 1$ because $c \leq 1$), i.e.

$$\frac{c}{3} \cdot t(x) \geq |x|^2 \qquad \text{and} \qquad \frac{1}{n^2} \cdot t(x) \geq \frac{|x|^2}{n^2}.$$

Similarly,

$$\frac{c}{3} \cdot t(x) \geq |f(x)|^2 \qquad \text{and} \qquad \frac{1}{n^2} \cdot t(x) \geq \frac{|f(x)|^2}{n^2}.$$

We thus want

$$\frac{c}{3} + \frac{1}{n^2} + \frac{8}{n} + \frac{c}{3} + \frac{1}{n^2} \leq c,$$

i.e.

$$\frac{2}{n^2} + \frac{8}{n} \leq \frac{c}{3},$$

and this determines an appropriate value for n. $\square$

The reference to the size of the output can be eliminated, if we restrict our attention to 0,1-valued functions.

Corollary VII.4.10 *If $c > 0$ is a real constant, then*

$$\lim_{x\to\infty} \frac{t(x)}{|x|^2} = \infty \;\Rightarrow\; \mathrm{TIME}\,[\,t\,] = \mathrm{TIME}\,[\,c\cdot t\,].$$

As for Space, the Linear Speed-Up Theorem for Time should also not be taken literally as saying that the time needed for a computation can always be (linearly) compressed by using a different machine. Again, it should be seen as showing that even in the abstract context of Turing machines, a notion of a time measure that accounts only for the number of moves made in a computation (and not for the number of states) is quite unrealistic: even by calling 'n moves of a given machine' a 'single move of a new machine', if the technological limits of state potentiality are reached, then making a single move of the new machine would still require n atomic acts.

Exercises VII.4.11 Time Speed-Up for Multitape Turing Machines. We extend Definition VII.4.1 to n-tape ($n \geq 2$) or multitape Turing machines, in the natural way.

a) *For n-tape ($n \geq 2$) or multitape Turing machines, if $c > 0$ is a real constant, then*

$$\lim_{x\to\infty} \frac{t(x)}{\max\{|x|, |f(x)|\}} = \infty \wedge f \in \mathrm{TIMEF}\,[\,t\,] \;\Rightarrow\; f \in \mathrm{TIMEF}\,[\,c\cdot t\,].$$

(Hartmanis and Stearns [1965]) (Hint: only a time $|x| + \frac{|x|}{n}$ is needed to translate an input of length $|x|$, since one can work out the translation on a working tape, and then proceed on this as if it were the input tape, while using the original input tape as a working tape. Similarly for the translation of the output.)

b) *For n-tape ($n \geq 2$) or multitape Turing machines, if $c > 0$ is a real constant, then*

$$\lim_{x\to\infty} \frac{t(x)}{|x|} = \infty \;\Rightarrow\; \mathrm{TIME}\,[\,t\,] = \mathrm{TIME}\,[\,c\cdot t\,].$$

Rosenberg [1967] has shown that the exercises above are the best possible, since *for n-tape ($n \geq 2$) or multitape Turing machines,* $\mathrm{TIME}\,[\,|x|\,] \neq \mathrm{TIME}\,[\,2\cdot|x|\,]$.

The Linear Speed-Up Theorems just proved are in some respects weaker and in other respects stronger than the Speed-Up Theorem VII.2.16. While the latter shows in a *nonconstructive* way that for *any* measure *some* function admits an *arbitrarily large* speed-up, the former show in a *constructive* way that practically *all* functions admit a *linear* speed-up w.r.t. *space and time*.

Improvements of the Linear Speed-Up Theorems are in Fischer [1967] and Moran [1981].

Hierarchy theorems

We now consider the effect of the relative asymptotic behavior of resource bounds on the set-theoretical inclusion of the complexity classes defined by them.

Proposition VII.4.12 *If s_1 and s_2 have the same asymptotic behavior, then they define the same space complexity class. More precisely,*

$$0 < \lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} < \infty \ \Rightarrow\ \mathrm{SPACEF}\,[\,s_1\,] = \mathrm{SPACEF}\,[\,s_2\,].$$

Proof. If $\lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} > 0$, then $\frac{s_1(x)}{s_2(x)} \geq c$ for some constant $c > 0$ and almost every x, and hence $c \cdot s_2(x) \leq s_1(x)$, i.e. $\mathrm{SPACEF}\,[\,c\cdot s_2\,] \subseteq \mathrm{SPACEF}\,[\,s_1\,]$. By the Space Linear Speed-Up Theorem, $\mathrm{SPACEF}\,[\,s_2\,] \subseteq \mathrm{SPACEF}\,[\,s_1\,]$.

Similarly, if $\lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} < \infty$, then $\mathrm{SPACEF}\,[\,s_1\,] \subseteq \mathrm{SPACEF}\,[\,s_2\,]$. Thus $\mathrm{SPACEF}\,[\,s_1\,] = \mathrm{SPACEF}\,[\,s_2\,]$. □

It remains to see what happens if s_1 have different asymptotic behavior. We only consider the condition $\lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} = 0$, because the symmetric condition $\lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} = \infty$ is symmetric.

If $\lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} = 0$, then $\frac{s_1(x)}{s_2(x)} \leq 1$ and hence $s_1(x) \leq s_2(x)$, for almost all x. Thus $\mathrm{SPACEF}\,[\,s_1\,] \subseteq \mathrm{SPACEF}\,[\,s_2\,]$. However, by the Downward Gap Theorem VII.3.13, there is no hope of proving $\mathrm{SPACEF}\,[\,s_1\,] \subset \mathrm{SPACEF}\,[\,s_2\,]$ in general. Thus some additional hypothesis is necessary.

From previous results, one might expect that what fails in general holds for step-counting functions. The next result shows that, indeed, *if a function is space constructible, then any function lying sufficiently below it names a different space complexity class.*

Theorem VII.4.13 Space Hierarchy Theorem (Hartmanis, Lewis and Stearns [1965]) *If s_2 is space constructible and $s_2(x) \geq \log|x|$, then*

$$\lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} = 0 \ \Rightarrow\ \mathrm{SPACE}\,[\,s_1\,] \subset \mathrm{SPACE}\,[\,s_2\,].$$

Proof. To show $\mathrm{SPACE}\,[\,s_1\,] \neq \mathrm{SPACE}\,[\,s_2\,]$, we build a 0,1-valued function f that diagonalizes against every Turing machine working in at most s_1 space, and use the hypothesis to show that f can be computed is s_2 space.

We let $\{\varphi_e\}_{e\in\omega}$ be an acceptable system of indices obtained by enumerating each Turing machine infinitely often as in VII.4.2, and let M_e be the Turing machine computing φ_e. We try to diagonalize against M_x by letting

$$f(x) = 1 - \varphi_x(x)$$

if the computation of $\varphi_x(x)$ does not take more than $s_2(x)$ space. Then, automatically, $f \in \text{SPACE}\,[\,s_2\,]$.

The construction of a Turing machine M computing f is as follows. On input x:

- First, M simulates a Turing machine that on input x takes exactly $s_2(x)$ space to halt (that such a machine exists follows from the hypothesis of space constructibility of s_2), and marks the boundary of the space so used.

- Second, M simulates M_x on input x and if the simulation halts in less than the marked space, then M outputs
$$f(x) = 1 - \varphi_x(x).$$

- If the simulation of M_x tries to step out of the marked space, or enters a loop, then M outputs
$$f(x) = 0.$$

To show that $f \in \text{SPACE}\,[\,s_2\,]$, the only delicate point is the detection of loops (in the last case of the definition of f) in space s_2. If we let f be undefined in the case of loops, then the associated machine would indeed require at most space $s_2(x)$ on input x. By VII.4.5, the hypothesis $s_2(x) \geq \log|x|$ implies that this machine can have less than $c^{s_2(x)}$ distinct configurations on input x, for some constant c, and hence it loops exactly if it makes at least $c^{s_2(x)}$ moves on input x. Using the space constructibility of s_2 again, one can thus define a new machine that supplements the old one, by adding a counter of length $s_2(x)$ (e.g. on an additional tape, whose addition is allowed by VII.4.6.1), and increasing it by 1 at each move of the old machine. By using base c, the counter thus allows us to count up to $c^{s_2(x)}$, and if this number of moves is reached, one knows a loop has been entered. One can then output $f(x) = 0$ as needed.

To show $f \notin \text{SPACE}\,[\,s_1\,]$, suppose $f \in \text{SPACE}\,[\,s_1\,]$. Then f is computed by a Turing machine M^* as in VII.4.2 taking at most space s_1, and M^* can be simulated by M in at most space $c \cdot s_1$ for some constant $c \geq 1$ (originated by the fact that M^* may have an arbitrary large alphabet, while M has a fixed alphabet). Since each Turing machine appears infinitely often in the given enumeration, and $\lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} = 0$ by hypothesis, there is an index x of M^* such that
$$c \cdot s_1(x) < s_2(x).$$

But then the construction of f would give
$$f(x) \simeq 1 - \varphi_x(x),$$

which is a contradiction (since x is an index of a machine M^* computing f, and hence $f \simeq \varphi_x$). $\Box$

Note that the hypothesis $s_2(x) \geq \log |x|$ was needed to halt, in space s_2, an undefined computation not requiring more than space s_2. Sipser [1980] has shown how to do this without the hypothesis, thus proving that *the Space Hierarchy Theorem holds under the only hypothesis of space constructibility of* s_2.

The two main tools used in the proof above are: a *simulation process* to diagonalize against given functions, and a *clock device* that shuts the computation off when the available resource has run out. For space complexity, the two do not interfere, since they can be done independently by first marking the space to be used, and then by working inside it.

For time complexity, complications occur from the need to do the two things simultaneously, and here the number of tapes becomes important:

- With multitape Turing machines, the clock is free (since it can run on a separate tape), but simulation is troublesome (since a loss factor is introduced by the fact that machines with any number of tapes have to be simulated by a machine with a fixed number of tapes).

- With a fixed number of tapes, simulation is easy (introducing only a constant factor of loss), but running the clock is troublesome (since the natural procedure would be to introduce an additional tape, whose elimination at the end introduces a loss factor).

These complications are reflected in the fact that a full analogue of the Space Hierarchy Theorem is not known to hold for time, for our model of a one-tape Turing machine (for other models, see the comments at the end of VII.4.16). Here we prove an approximation with a slightly weaker conclusion.

Exercise VII.4.14 *If t_2 is time constructible then, for some constant c,*

$$\lim_{x\to\infty} \frac{t_1(x)}{t_2(x)} = 0 \;\Rightarrow\; \mathrm{TIME}\,[\,t_1\,] \subset \mathrm{TIME}\,[\,c \cdot (t_2)^2\,].$$

(Hartmanis and Stearns [1965], Hartmanis [1968]) (Hint: proceed as in VII.4.13. The only difference is that we have to run simultaneously (in parallel) a Turing machine that on input x takes exactly time $t_2(x)$ to halt. Thus M needs two tapes, one to simulate this machine, and the other to simulate M_x on input x. The construction is as above, and it produces a function f such that $f \notin \mathrm{TIME}\,[\,t_1\,]$ as before. But we only obtain f computable in time t_2 by a two-tape Turing machine M, and thus computable in time $c \cdot (t_2)^2$ in the sense of VII.4.1, by VII.4.6.)

Theorem VII.4.15 Time Hierarchy Theorem (Hartmanis and Stearns [1965], Hartmanis [1968]) *If t_2 is time constructible then, for some constant c,*

$$\lim_{x\to\infty} \frac{t_1(x)}{t_2(x)} = 0 \;\Rightarrow\; \text{TIME}\,[\,t_1\,] \subset \text{TIME}\,[\,c \cdot t_2 \cdot \log t_2\,].$$

Proof. We use a Turing machine M with a single tape having three tracks: one with a counter of length $\log t_2(x)$, one for simulations, and one for the representation of the input x.

We first mark the boundaries of the counter, in time proportional to $t_2(x)$. This is possible because t_2 is time constructible, and thus there is a Turing machine that halts on input x in exactly $t_2(x)$ steps. One can then simulate such a machine, and keep a binary counter that increases by one at each move. When the simulation ends, the counter will contain $t_2(x)$ in binary representation, and it will thus have length $\log t_2(x)$.

At every move of our machine, we will keep the counter close to the head (this requires possible shifts of its content one cell to the right or the left, and hence a number of moves proportional to $\log t_2(x)$), and decrease it by one. Thus f is computed in time proportional to $t_2(x) \cdot \log t_2(x)$.

We also keep the input x close to the head, because we need knowledge of x to simulate M_x, and if x were not moved, we would have to go back to it when needed, losing control on the number of moves needed in the simulation. There is then a fixed constant c_x such that one move of M_x can be simulated by c_x moves of M (including those needed to shift x and decode it). Thus, if M_x works in time $t_1(x)$ on input x, M can simulate its behavior in time $c_x \cdot t_1(x)$. Since

$$\lim_{x\to\infty} \frac{t_1(x)}{t_2(x)} = 0,$$

and each machine is enumerated infinitely often, one can then proceed as in VII.4.13. □

Exercises VII.4.16 Time Hierarchies for Multitape Turing Machines. We extend Definition VII.4.1 to n-tape ($n \geq 2$) or multitape Turing machines, in the natural way.

a) *For n-tape ($n \geq 2$) or multitape Turing machines, if t_2 is time constructible, then for some constant c*

$$\lim_{x\to\infty} \frac{t_1(x)}{t_2(x)} = 0 \;\Rightarrow\; \text{TIME}\,[\,t_1\,] \subset \text{TIME}\,[\,c \cdot t_2 \cdot \log t_2\,].$$

(Hint: as in VII.4.14, using the simulation result VII.4.7.b.)

b) *For n-tape ($n \geq 2$) or multitape Turing machines, if t_2 is time constructible, then*

$$\lim_{x\to\infty} \frac{t_1(x) \cdot \log t_1(x)}{t_2(x)} = 0 \;\Rightarrow\; \text{TIME}\,[\,t_1\,] \subset \text{TIME}\,[\,t_2\,].$$

(Hartmanis and Stearns [1965], Hennie and Stearns [1966]) (Hint: having at least three tapes, one can be used for clockwork, and the other two to simulate any multi-tape Turing machine, with at most a logarithmic time loss by VII.4.7.b.)

Fürer [1982] has shown that the exercises above are not the best possible, since *for n-tape ($n \geq 2$) Turing machines, if t_2 is time-constructible, then*

$$\lim_{x\to\infty} \frac{t_1(x)}{t_2(x)} = 0 \ \Rightarrow\ \text{TIME}\,[\,t_1\,] \subset \text{TIME}\,[\,t_2\,].$$

Thus, for n-tape ($n \geq 2$) Turing machines, a full analogue of the Space Hierarchy Theorem VII.4.13 holds for Time.

For other results see Hartmanis [1968], [1979], Strnad [1968], Hopcroft and Ullman [1969], [1979], Glinert [1971], Seiferas [1977], [1977a], Paul [1979], Sipser [1980], Fürer [1982], Zak [1983].

Space versus time

Until now, we have examined space and time complexities in parallel, by proving similar results for them. We turn now to their mutual relationships.

Theorem VII.4.17 Space-Time Trade-Off Theorem.

1. *A function computable in time t is computable in space t, i.e.*

$$\text{TIMEF}\,[\,t\,] \subseteq \text{SPACEF}\,[\,t\,].$$

2. *If $s(x) \geq \log|x|$, then a function computable in space s is computable in time c^s for some constant c, i.e.*

$$\text{SPACEF}\,[\,s\,] \subseteq \bigcup_{c\in\omega} \text{TIMEF}\,[\,c^s\,] = \bigcup_{c\in\omega} \text{TIMEF}\,[\,2^{c\cdot s}\,].$$

Proof. Part 1 is trivial, since every move of a Turing machine uses at most one new cell of the tape, and thus a computation does not take more space than time.

For part 2, let f be a recursive function computed by a Turing machine M in space $s(x)$. Since M always halts, it cannot loop on any input x, i.e. no configuration can be repeated twice during a computation on x. Thus the number of possible configurations on input x, on a tape of length $s(x)$, bounds the time needed for the computation. By VII.4.5, f is then computable in time c^s for some c. □

From the previous result, it follows that *any class of functions closed under exponentiation with constant bases provides the same bounds for space and time* *complexity measures*. For example, this holds for the elementary or the primitive recursive functions (see VIII.7.6 and VIII.8.8), but it may fail for smaller classes. For example, it is not known whether the functions computable in polynomial space (in the length of the input) are also computable in polynomial time (in the length of the input). The relative classes are called PSPACEF and PF, and are considered in Sections VIII.5 and VIII.2.

The problem just hinted at is clearly connected to the possibility of improving the exponential loss in VII.4.17.2. At present, this is the best known result. On the other hand, various improvements of VII.4.17.1 are known, and show that *space is a resource strictly better than time*. For example:

- for one-tape Turing machines, if t is time constructible and $t(x) \geq |x|$, then
$$\text{TIMEF}\,[\,t^2\,] \subseteq \text{SPACEF}\,[\,t\,]$$
(Paterson [1972], see also Ibarra and Moran [1983], Liskiewicz and Lorys [1989])
- for n-tape ($n \geq 2$) or multitape Turing machines,
$$\text{TIMEF}\,[\,t \cdot \log t\,] \subseteq \text{SPACEF}\,[\,t\,]$$
(Hopcroft, Paul and Valiant [1977], see also Adleman and Loui [1981]).

It follows that

$$s \textit{ space constructible and } s(x) \geq x \;\Rightarrow\; \text{TIME}\,[\,s\,] \subset \text{SPACE}\,[\,s\,].$$

Indeed, by taking $t = \sqrt{s}$, one obtains $\text{TIMEF}\,[\,s\,] \subseteq \text{SPACEF}\,[\,\sqrt{s}\,]$ from the results quoted above. By using the hypothesis on s one has $s(x) \leq (s(x))^2$, hence $\lim_{x\to\infty} \frac{\sqrt{s(x)}}{s(x)} = 0$, and $\text{SPACE}\,[\,\sqrt{s}\,] \subset \text{SPACE}\,[\,s\,]$ by the Space Hierarchy Theorem.[3]

Some additional hypothesis on s is necessary, since the result fails in general. Indeed, by the Downward Gap Theorem VII.3.13 and for some constant c,

$$\text{TIMEF}\,[\,s\,] \subseteq \text{SPACEF}\,[\,s\,] \subseteq \text{TIMEF}\,[\,2^{c\cdot s}\,] \subseteq \text{TIMEF}\,[\,2^{2^s}\,],$$

and thus it is enough to consider a time gap of doubly exponential width to obtain a function s such that $\text{TIMEF}\,[\,s\,] = \text{SPACEF}\,[\,s\,]$.

For more space-time trade-off results, see Hopcroft and Ullman [1968], Paul and Reischuck [1981], Chandra, Kozen and Stockmeyer [1981], Halpern, Loui, Meyer and Weise [1986].

[3]Hartmanis, Chang, Chari, Ranjan and Rohatgi [1992] have proved that the result does *not* relativize, since there is an oracle A s.t. $\text{TIME}^A[\,s\,] = \text{SPACE}^A[\,s\,]$ for any s. Thus, even a result formalizing a basic intuition about computations, namely that space is better than time, may not extend to computations relative to an oracle.

Nondeterministic Turing machines

Recall (from Section I.4) that a **nondeterministic Turing machine** is simply a Turing machine whose instructions are not required to be consistent, so that more than one could be applicable in a given situation. In particular, there is possibly more than one acceptable behavior, and hence more than one output, on a given input. This raises a problem that can be approached in any of the following ways:

- By seeing nondeterministic Turing machines as devices computing *many-valued functions*.

  This would distract us from our basic interest in usual one-valued functions.

- By imposing an additional *one-valuedness condition*, requiring all possible outputs on a given input to be the same.

  This is not satisfactory from a computational point of view, since it requires a check on all possible computations on a given input, thus spoiling the possibility of having an *Enumeration Theorem for nondeterministic Turing machines* (a similar problem arose in Section II.3 for partial recursive operators).

- By restricting attention to *partial functions* with 1 as the only input, and considering as undefined all computations not giving 1 as output.

  This has the advantages of the previous approach (one-valuedness being ensured automatically), without its disadvantages. Of course, this approach restricts the attention to sets (as opposed to functions), with an emphasis on membership (as opposed to full characteristic functions). It is thus particularly well suited for studying analogues of r.e. sets.

In the following, when dealing with nondeterminism we will always restrict ourselves to the last option, and will call *accepting computation* any computation giving 1 as output.

Definition VII.4.18 Nondeterministic Acceptance of Sets (Rabin and Scott [1959])

1. *A set A is* **acceptable in nondeterministic time t** *if there is a nondeterministic Turing machine as in VII.4.1 such that, for any x, $x \in A$ if and only if there is an accepting computation on input x not requiring more than $t(x)$ moves.*

   We denote by **NTIME** $[\,\boldsymbol{t}\,]$ *the class of sets acceptable in nondeterministic time t.*

2. *A set A is* **acceptable in nondeterministic space s** *if there is a nondeterministic Turing machine as in VII.4.2 such that, for any x, $x \in A$ if and only if there is an accepting computation on input x not requiring more than $s(x)$ cells of the working tape.*

   We denote by **NSPACE** $[\,\boldsymbol{s}\,]$ *the class of sets acceptable in nondeterministic space s.*

Thus $t(x)$ and $s(x)$ are bounds on the time and space needed to obtain the output 1 on input x when $x \in A$, in *some* way. In particular, a nondeterministic Turing machine accepting a set can still have nonterminating computations, or computations that require more than the given time or space bounds. It is natural to try to disregard (in the sense of turning off) computations that take more than $t(x)$ time or $s(x)$ space on input x. This can be done at no cost if s is space constructible, and at the cost of adding a clock (i.e. an additional tape) if t is time constructible.

Simulation of nondeterministic multitape Turing machines by nondeterministic *one-tape* Turing machines can be done with *no space loss* and a *quadratic time loss*, as in VII.4.6. Moreover, simulation of nondeterministic multitape Turing machines by nondeterministic *two-tape* Turing machines can be done with *no time loss*, in contrast to the logarithmic time loss provided by VII.4.7.b for deterministic Turing machines.

Exercise VII.4.19 *A set acceptable in time t by a nondeterministic multitape Turing machine is acceptable in time t by a nondeterministic two-tape Turing machine.* (Book, Greibach and Wegbreit [1970]) (Hint: given an n-tape nondeterministic Turing machine M, we build a two-tape nondeterministic Turing machine M_1 as follows. The first tape of M_1 consists of n double tracks simulating the tapes and head positions of M, as in VII.4.6. The second tape of M_1 consists of multiple tracks simulating a computation of M, i.e. the scanned cell represents the current situation and the current instruction. To simulate M, M_1 first guesses a whole computation on the second tape, and then checks whether each double track of the first tape confirms the guess of the second tape. Each check takes at most $t(x)$ steps in one direction and $t(x)$ to come back, so the whole simulation takes at most $2n \cdot t(x)$ steps. By an analogue of the Linear Speed-Up Theorem for Time, this can be reduced to $t(x)$ steps.)

As we have already briefly mentioned, nondeterministic acceptance by a Turing machine bears similarity to the process of determining membership to an r.e. set. In both cases, an input is accepted if *some* short computation accepts it, while it is rejected if *no* short computation accepts it, where a computation is short if it does not exceed the allowed resource bound (which, in the case of r.e. sets, is simply a finiteness requirement).

One could expect that, in general, determination of *non*membership would exceed the allowed resource bounds. For example, the existence of r.e. nonrecursive sets shows that determination of nonmembership to an r.e. set may require infinitely long computations.

The exact determination of the loss involved in the step from nondeterministic computations (for membership alone) to deterministic ones (for both membership and nonmembership) is a central concern of Complexity Theory, with the following puzzling aspect: naive simulations provide trivial upper bounds which, in many cases, have been neither improved nor proved to be optimal.

The next result is an exception, and shows how to simulate nondeterministic space bounded computations in a deterministic way, with only a quadratic loss. The crucial property needed in the proof is that space is a reusable resource.

Theorem VII.4.20 Savitch's Theorem (Savitch [1970], Tseitin) *If s is space constructible and $s(x) \geq \log|x|$, a set acceptable in nondeterministic space s is computable in deterministic space s^2, i.e.*

$$\text{NSPACE}\,[\,s\,] \subseteq \text{SPACE}\,[\,s^2\,].$$

Proof. Suppose A is accepted by a nondeterministic Turing machine N working in space s. We want to build a deterministic Turing machine M working in space s^2 and computing A.

Since N works in space $s(x) \geq \log|x|$, there are only $2^{a \cdot s(x)}$ possible different configurations on input x, for some a (because, by VII.4.5, there are only

$$d^{s(x)} = (2^{\log d})^{s(x)} = 2^{(\log d) \cdot s(x)}$$

such possible configurations, for some d). By possibly modifying N in an inessential way, we can suppose that on any input x there are unique initial and final configurations c_0^x and c_f^x. For example, after N reaches any accepting configuration, its work can be continued by erasing the working tape, moving the input head to a fixed position (e.g. to the rightmost symbol of the input), and then entering a unique special final state.

Suppose c_1 and c_2 are two configurations of N. We let

$$c_1 \vdash_n c_2 \;\Leftrightarrow\; c_2 \text{ can be obtained from } c_1 \text{ in at most } 2^n \text{ moves of } N.$$

Since if there is a computation of N accepting x, then there is one of minimal length, in particular one in which no configuration is repeated twice, to determine whether $x \in A$ it suffices to determine whether

$$c_0^x \vdash_{a \cdot s(x)} c_f^x,$$

and we are thus reduced to the problem of finding a deterministic Turing machine M computing $\vdash_{a \cdot s(x)}$ in space proportional to $s(x)$, uniformly in x.

By induction on n, we show how to compute $\vdash_n$ in a deterministic space proportional to $n \cdot s(x)$, uniformly in n. In particular, $\vdash_{a \cdot s(x)}$ is then computable in deterministic space proportional to $s^2(x)$, and hence (by the Space Linear Speed-Up Theorem VII.4.8) in deterministic space $s^2(x)$, as required.

For $n = 0$, $c_1 \vdash_0 c_2$ if and only if $c_1 = c_2$, or c_2 can be obtained from c_1 in one move of N (since $2^0 = 1$).

For $n + 1$, suppose we know how to compute $\vdash_n$ in deterministic space proportional to $n \cdot s(x)$. Notice that

$$c_1 \vdash_{n+1} c_2 \Leftrightarrow (\exists c)(c_1 \vdash_n c \wedge c \vdash_n c_2).$$

Indeed, we can take as c a configuration half-way between c_1 and c_2, i.e. obtained after 2^n steps of the transformation of c_1 into c_2 (which takes at most 2^{n+1} steps). Then $\vdash_{n+1}$ is computable in a deterministic space proportional to $(n + 1) \cdot s(x)$, as follows:

- We successively generate all possible configurations c on input x. This can be done in space proportional to $s(x)$, since any such configuration consists of a description of the working tape (which takes space $s(x)$), the state and the position of the working head (the state can be inserted at the appropriate place in the description of the working tape, and thus takes only constant additional space), and the position of the input head (which can be described by a number $\leq |x|$ and hence be written, using binary notation, in space $\log |x| \leq s(x)$).

  Since s and log are space constructible, the space needed to write any such c down can be marked ahead of time, and thus one only needs to generate the configurations of that allowed length.

- For each such configuration c, we successively check whether $c_1 \vdash_n c$ and $c \vdash_n c_2$. Each check requires space proportional to $n \cdot s(x)$ by inductive hypothesis, and can be done always using the same space (because space is reusable).

Since the space required is proportional to $s(x)$ in the first part and to $n \cdot s(x)$ in the second, the total amount of space needed is proportional to $(n+1) \cdot s(x)$.

Summing up the inductive procedure, a typical space needed to check whether $c_1 \vdash_n c_2$ is a sequence of n blocks of the form

$$\underbrace{c_1 * c_2 * n * c}_{n} * \underbrace{c_1 * c * n - 1 * c'}_{n-1} * \cdots,$$

each of length linear in $s(x)$. The whole procedure thus takes space proportional to $n \cdot s(x)$. $\square$

From the previous result it follows that *any class of sufficiently large space constructible functions closed under squaring provides the same bounds for deterministic and nondeterministic space complexity measures*. For example, this holds for the polynomials, whose relative class of functions is called PSPACEF and is considered in Section VIII.5, but it may fail for smaller classes. For example, it is not known whether the sets acceptable in nondeterministic logarithmic space (in the length of the input) are also computable in deterministic logarithmic space (in the length of the input). The relative classes are called NLOGSPACE and LOGSPACE, and are considered in Section VIII.1. Similarly for nondeterministic and deterministic linear space, whose relative classes are called NLINSPACE and LINSPACE and are considered in Section VIII.5.

The problem just referred to is clearly connected to the possibility of improving the quadratic loss in VII.4.20, which at present is the best known result. See Tompa [1981] and Monien and Sudbourough [1982] for a complementary result, showing that *if s is space constructible and* $\log\log|x| \leq s(x) < \log|x|$, *then* $\text{NSPACE}[\,s\,] \subseteq \text{SPACE}[\,s \cdot \log s\,]$, and Chandra, Kozen and Stockmeyer [1981] for an improved result, showing that $\text{NSPACE}[\,s\,] \subset \text{SPACE}[\,s^2\,]$.

As for nondeterministic *time* bounds, the only known result is the following trivial one.

Proposition VII.4.21 *A set acceptable in nondeterministic time t is computable in deterministic time c^t for some constant c, i.e.*

$$\text{NTIME}[\,t\,] \subseteq \bigcup_{c \in \omega} \text{TIME}[\,c^t\,].$$

Proof. Suppose N is a nondeterministic Turing machine accepting a set A in time t. For each input x, consider the tree of all computations consisting of at most $t(x)$ moves. By definition of acceptance in nondeterministic time t, $x \in A$ if and only if at least one branch of this tree accepts.

To compute A deterministically, a Turing machine has only to check all possible branches of this tree, and stop as soon as one of these accepts. But the tree is finitely branching (since N has only a finite number of instructions, each of which determines a possible move, and hence a possible new node, at any instant), and each branch has at most length $t(x)$. Thus the whole tree has at most $c^{t(x)}$ nodes, for some constant c.

A deterministic Turing machine M that, on input x, systematically simulates all possible moves of N, will then decide whether $x \in A$ in a number of moves proportional to $c^{t(x)}$, for some c. $\square$

From the previous result it follows that *any class of functions closed under exponentiation with constant bases provides the same bounds for deterministic and nondeterministic time complexity measures*. For example, this holds for the elementary or the primitive recursive functions (see VIII.7.6 and VIII.8.8), but it may fail for smaller classes. For example, it is not known whether the sets acceptable in nondeterministic polynomial time (in the length of the input) are also computable in deterministic polynomial time (in the length of the input). The relative classes are called NP and P, and are considered in Sections VIII.3 and VIII.2.

Savitch's Theorem determines the space loss needed to turn a nondeterministic procedure to accept a set into a deterministic procedure to compute it, and it constitutes a complexity-theoretical analogue of the recursion-theoretical determination of an oracle in which a given r.e. set is recursive.

We turn now to the related problem of *closure under complement* of a complexity class. From a recursion-theoretical point of view, the recursive sets are closed under complement, while the r.e. sets are not. From a complexity-theoretical point of view, we might thus guess that the deterministic complexity classes are closed under complement, while the nondeterministic complexity classes are not. The first guess is confirmed by the next exercises, but the second guess is disproved by the next theorem.

Exercises VII.4.22 a) TIME $[\,t\,]$ *is closed under complement.* (Hint: switch accepting and rejecting states.)

b) *If s is space constructible and $s(x) \geq \log|x|$,* SPACE $[\,s\,]$ *is closed under complement.* (Hint: under the hypothesis, one can detect loops in space s.)

By VII.4.20 most of the nondeterministic space complexity classes coincide with the corresponding deterministic space complexity classes and are thus closed under complement. The gap for small space bounds left open by VII.4.20 is closed by the next result.

Theorem VII.4.23 Immerman and Szelepcsényi's Theorem (Immerman [1988], Szelepcsényi [1988]) *If s is space constructible and $s(x) \geq \log|x|$, then* NSPACE $[\,s\,]$ *is closed under complement.*

Proof. Suppose A is accepted by a nondeterministic Turing machine N working in space s. We want to build a nondeterministic Turing machine N_1 working in space s and accepting $\overline{A}$.

Since N works in space $s(x) \geq \log|x|$, by VII.4.5 there are only $d^{s(x)}$ possible different configurations on input x, for some d. By possibly modifying N in an inessential way, we can suppose that on any input x there are unique initial and final configurations c_0^x and c_f^x. For example, after N reaches any accepting configuration, its work can be continued by erasing the working tape, moving the input head to a fixed position (e.g. to the rightmost symbol of the input), and then entering a unique special final state.

If we let

$$C_n^x = \{c : c \text{ is a configuration reachable in at most } n \text{ steps from } c_0^x\},$$

then

$$x \in A \;\Leftrightarrow\; c_f^x \in C^x_{d^{s(x)}},$$

and hence

$$x \in \overline{A} \;\Leftrightarrow\; c_f^x \in \overline{C^x_{d^{s(x)}}},$$

because if there is a computation of N accepting x, then there is one of minimal length, in particular one in which no configuration is repeated twice.

We are thus reduced to the problem of finding a nondeterministic Turing machine N_1 accepting, on input x, the complement of $C^x_{d^{s(x)}}$. Note that if we knew

1. how to accept $C^x_{d^{s(x)}}$
2. how to compute $|C^x_{d^{s(x)}}|$, i.e. how many elements are in $C^x_{d^{s(x)}}$

in nondeterministic space $s(x)$, then we could also accept $\overline{C^x_{d^{s(x)}}}$ in nondeterministic space $s(x)$, as follows.[4] Given a configuration, we first guess $|C^x_{d^{s(x)}}|$ distinct configurations on input x, which can be done in nondeterministic space $s(x)$ by hypothesis 2. Then we check whether all guessed configurations are in $C^x_{d^{s(x)}}$, which can be done in nondeterministic space $s(x)$ by hypothesis 1. By comparing the given configuration with the guessed ones, we can determine in nondeterministic space $s(x)$ whether it is in $C^x_{d^{s(x)}}$ or not, i.e. whether it is in $C^x_{d^{s(x)}}$ or in $\overline{C^x_{d^{s(x)}}}$.

To prove that $\overline{C^x_{d^{s(x)}}}$ is acceptable in nondeterministic space $s(x)$, it is thus enough to show, for any n,

1. how to accept C_n^x
2. how to compute $|C_n^x|$

in nondeterministic space $s(x)$. We proceed by induction on n:

- For $n = 0$, there is only one configuration in C_n^x, namely the initial configuration c_0^x, and hence $|C_n^x| = 1$.

[4]This is a complexity-theoretical analogue of the trivial recursion-theoretical fact that, if one knows not only an r.e. index of a finite set D, but also its cardinality $|D|$, then one can (effectively) obtain a characteristic (or even a canonical) index of D, by just enumerating the set D until exactly $|D|$ elements have been found.

The additional information on $|D|$ is essential, since (by II.5.14) the result fails without it.

- For $n+1$, we first guess $|C_n^x|$ distinct configurations on input x, which can be done in nondeterministic space $s(x)$ by hypothesis 2. Then we check whether all guessed configurations are in C_n^x, which can be done in nondeterministic space $s(x)$ by hypothesis 1. By cycling through all possible configurations on input x, and seeing which ones follow from any of the guessed configurations in a single step, we both accept C_{n+1}^x and determine $|C_{n+1}^x|$ in nondeterministic space $s(x)$.[5] □

From the previous result it follows that *any nondeterministic space complexity class defined by sufficiently large bounds is closed under complement.* For example, this holds for the classes NLOGSPACE and NLINSPACE considered in Sections VIII.1 and VIII.5. Nothing is known instead for nondeterministic *time* complexity classes, such as the class NP considered in Section VIII.3.

As we have already noted in the proof above, *nondeterministic versions of the Space and Time Linear Speed-Up Theorems hold*, by the same proofs. Book and Greibach [1970] have shown that such results are the best possible.

The next result provides a nondeterministic version of VII.4.13.

Theorem VII.4.24 Nondeterministic Space Hierarchy Theorem (Ibarra [1972], Book [1976], Seiferas [1977a], Fischer) *If s_2 is space constructible and $s_2(x) \geq \log|x|$, then*

$$\lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} = 0 \Rightarrow \text{NSPACE}[\,s_1\,] \subset \text{NSPACE}[\,s_2\,].$$

Proof. We show how to modify the proof of VII.4.13. The only delicate point in that proof was the detection of loops, which can be dealt with here by taking advantage of the proof of VII.4.23. More precisely, let $\{N_e\}_{e\in\omega}$ be an effective enumeration of the nondeterministic Turing machines, in which each machine appears infinitely often. For each nondeterministic Turing machine N_e, let N'_e be a nondeterministic Turing machine (effectively obtained as in the proof of VII.4.23) such that N'_e accepts x in space $s_2(x)$ if and only if N_e does not accept x in space $s_2(x)$.

We construct a Turing machine N as follows. On input x:

- First, N simulates a Turing machine that on input x takes exactly $s_2(x)$ space to halt (that such a machine exists follows from the hypothesis of space constructibility of s_2), and marks the boundary of the space so used.

[5] We cannot simply compute $|C_{n+1}^x|$ by generating the configurations in C_n^x and making all possible single moves from them, because there might be configurations in C_{n+1}^x that can be obtained in more than one way from configurations in C_n^x, and they would thus be counted more than once.

- Second, N simulates N'_x on input x and if the simulation halts in less than the marked space, then N accepts x if and only if N'_x does (i.e. if and only if N_x does not accept x).

By construction, N works in space s_2. As in VII.4.13, N diagonalizes against every nondeterministic Turing machines working in space s_1. $\square$

Nondeterministic versions of the Time Hierarchy Theorem VII.4.15 have been proved by Cook [1973], Seiferas, Fischer and Meyer [1978], Zak [1983], Li [1985] (see Hopcroft and Ullman [1979], and Wagner and Wechsung [1986]).

The next result provides a nondeterministic version of VII.4.17.

Theorem VII.4.25 Nondeterministic Space-Time Trade-Off Theorem (Cook [1971a])

1. *A set acceptable in nondeterministic time t is computable in deterministic space t, i.e.*
$$\mathrm{NTIME}\,[\,t\,] \subseteq \mathrm{SPACE}\,[\,t\,].$$
2. *If s is space constructible and $s(x) \geq \log|x|$, then a set acceptable in nondeterministic space s is computable in deterministic time c^s for some constant c, i.e.*

$$\mathrm{NSPACE}\,[\,s\,] \subseteq \bigcup_{c\in\omega} \mathrm{TIME}\,[\,c^s\,] = \bigcup_{c\in\omega} \mathrm{TIME}\,[\,2^{c\cdot s}\,].$$

Proof. Part 1 is trivial, since the simulation of a *single* nondeterministic computation requiring time t takes only space t, and space can be reused. Thus simulation of *all* nondeterministic computations can be done in the same space.

To prove Part 2, let M be a nondeterministic Turing machine M working in space s and accepting a set A. To compute A in deterministic time we successively write down, starting from the initial configuration on input x, all new configurations that can be obtained from previously written configurations by a move of M. When no new configuration can be thus obtained, we have written down all possible configurations that can be reached from the initial one, and it is enough to check whether one of them corresponds to an accepting computation, i.e. whether there is one such configuration with a halting accepting state.

By VII.4.5, there is a constant c such that there are at most $c^{s(x)}$ possible different configurations of M on input x, and each of them has length proportional to $s(x)$. Thus, at any stage we have at most $c^{s(x)}$ configurations already written down on a first tape, and each of them generates (in a single move) at most n ones (where n is the number of instructions of M), that can be written on a second tape. For each of the configurations on the second tape we have to check whether it is new, i.e. whether it is not already on the first tape. But to compare two given configurations on different tapes requires a time proportional to their length $s(x)$, and thus to compare all given configuration on the second tape with all the ones on the first tape requires time proportional $c^{s(x)}$. Thus the whole process takes a time proportional to $c^{s(x)}$, and by the Time Speed-Up Theorem A is computable in time $c^{s(x)}$. $\square$

For more nondeterministic space-time trade-off results, see Book, Greibach and Wegbreit [1970], Cook [1971a], and Paterson [1972].

Alternating Turing machines $\star$

The notion of parallel computation has been formalized in the previous subsection by means of a *software* device, i.e. by allowing the program of a Turing machine to contain incompatible instructions that could apply in a given situation. We now examine an alternative formalization of parallelism obtained by means of a *hardware* device, introduced and studied by Chandra, Kozen and Stockmeyer [1981].

An *alternating Turing machine* is a nondeterministic Turing machine that has two possible modes of computation, among which it can switch during a computation. The idea is that the machine accepts an input if either it is in an *existential* mode and some of its computations is accepting, or it is in a *universal* mode and all of its computations are accepting. To keep track of the possible switch of modes during a computation one proceeds inductively, and defines a node of the tree of all possible computations to be accepting if either it is existential and some immediate successor accepts, or it is universal and all immediate successors accept. Then an input is accepted if and only if the initial node of the computation tree accepts it.

The alternating complexity classes ASPACE $[s]$ and ATIME $[t]$ are defined in the natural way and are machine-independent, in the sense that the same classes are obtained by considering one-tape, n-tape ($n \geq 2$) or multitape alternating Turing machines (for nondeterministic Turing machines there is a time difference only between one and two tapes, by VII.4.19, while for deterministic Turing machines there is a time difference between n and $n+1$ tapes for any $n \geq 1$, by an extension of VII.4.7.b).

The *Linear Speed-Up Theorems* hold for alternating space and time, i.e. ASPACE $[s]$ = ASPACE $[c \cdot s]$ and ATIME $[t]$ = ATIME $[c \cdot t]$, for any constant $c > 0$.

The *Hierarchy Theorems* hold for alternating space and time, i.e. if s_2 is space constructible and $s_2(x) \geq \log|x|$, then

$$\lim_{x\to\infty} \frac{s_1(x)}{s_2(x)} = 0 \ \Rightarrow\ \text{ASPACE}\,[\,s_1\,] \subset \text{ASPACE}\,[\,s_2\,],$$

and if t_2 is time constructible, then

$$\lim_{x\to\infty} \frac{t_1(x)}{t_2(x)} = 0 \ \Rightarrow\ \text{ATIME}\,[\,t_1\,] \subset \text{ATIME}\,[\,t_2\,].$$

Closure under complement holds for alternating space and time, i.e. if s is space constructible and $s(x) \geq \log|x|$, then ASPACE$\,[\,s\,]$ is closed under complement, and if t is time constructible, then ATIME$\,[\,t\,]$ is closed under complement.

The *Space-Time Trade-Off Theorem* holds in the following form: if s is space constructible and $s(x) \geq \log|x|$, then

$$\text{NSPACE}\,[\,s\,] \subseteq \text{ATIME}\,[\,s^2\,] \qquad \text{and} \qquad \text{ASPACE}\,[\,s\,] \subseteq \bigcup_{c\in\omega} \text{TIME}\,[\,c^s\,],$$

and if t is time constructible and $t(x) \geq \log|x|$, then

$$\text{ATIME}\,[\,t\,] \subseteq \text{SPACE}\,[\,t^2\,].$$

From the Space-Time Trade-Off Theorem it follows that *any class of functions closed under squaring provides alternating space bounds one logarithm lower than the corresponding deterministic time bounds, and alternating time bounds equivalent to the corresponding deterministic space bounds*. For example, the class ALOGSPACE of sets acceptable by alternating Turing machines in logarithmic space is equal to the class P of sets computable by deterministic Turing machines in polynomial time, studied in Section VIII.2. Similarly, the class AP of sets acceptable by alternating Turing machines in polynomial time is equal to the class PSPACE of sets computable by deterministic Turing machines in polynomial space, studied in Section VIII.5. Finally, the class APSPACE of sets acceptable by alternating Turing machines in polynomial space is equal to the class POLYEXP of sets computable by deterministic Turing machines in polyexponential time, studied in Section VIII.6. Other alternating complexity classes are mentioned in Chapter VIII.

For a treatment of alternating Turing machines see Balcázar, Díaz and Gabarró [1990].

VII.5 Inductive Inference

In this section we turn our attention to the process of inductive inference, which can be described as a particular step from chaos (a sequence of accidents) to order (a pattern), or from effects (a sequence of events) to causes (a possible explanation of what produces them).

More precisely, in such a process we can isolate three distinct aspects:

- *data*
  They provide bits of information about the phenomena to be inferred, at given instants of time. According to whether the order in which they occur is considered to be relevant or not, the flow of data can be taken to be a *function* or a *set*. For example, the evolution of a physical system in time can be identified with a function, while a language can be identified with a set of strings.

  We deal with the first case in the present section, and with the second case in Section IX.5.

- *goals of inference*
  A theoretical goal is to be able to *explain* the phenomena to which the data refer, while a practical goal is to be able to *predict* the flow of data from a certain point on. In the first case, one is interested in understanding causes, in the second in reproducing effects (respectively, the tasks of *science* and *technique*).

  The various notions of inference dealt with in the present section, with the only exception of VII.5.1, will refer to the second case.

- *time needed for inference*
  An inference must be completed in a *finite* time, which could be *specified* in advance, *unspecified but recognizable*, or *unrecognizable*. In the first two cases one knows when the inference process has been completed. In the last two cases one knows that the process will be completed, but does not know in advance when.

  Since the first two cases of inference can occur only when the process to be inferred can be completely determined by a recognizable finite amount of data information, we will concentrate here on the third case (called limit inference).

As a first approximation to an inference process of the kind described above, we picture time as consisting of discrete intervals, and events as being codifiable by natural numbers. Thus a phenomenon to be inferred may be thought of as a function on the natural numbers. Any such function f is given by a sequence of values

$$f(0), \ldots, f(n), \ldots$$

The function can be inferred if this is not just a sequence of accidents, but rather it has an intrinsic necessity. We can specify this internal structure of the sequence of values in at least two ways, corresponding to the two goals described above:

- On the one hand, we can ask for a method that would allow us to predict the next value $f(n+1)$, once the values $f(0), \ldots, f(n)$ have been exhibited, for an arbitrary n.
- On the other hand, we can ask for a finite description that would compress the infinite amount of information contained in the sequence of values.

Even in this vague formulation, one can identify the functions which are (in principle) inferable w.r.t. any of the two methods with the recursive functions:

- On the one hand, since the values of a recursive function can be computed individually, they do not need knowledge of previous values. Conversely, by I.7.1, the class of recursive functions is closed under course-of-value recursion, i.e. an effective dependence of $f(n+1)$ on $f(0), \ldots, f(n)$.
- On the other hand, the functions with a program are exactly the recursive functions.

This completely disposes of the problem of which functions on natural numbers are *individually* inferable, and we can thus turn our attention to *classes* of functions. The problem here is that, for each class $\mathcal{C}$ of recursive functions, one could be able to infer members of $\mathcal{C}$ individually, without having a master inference method that would work uniformly for all members of $\mathcal{C}$.

Many possible formalizations of notions of inference for classes of total recursive functions have been considered in the literature. We confine ourselves here to a few of them, and refer to Klette and Wiehagen [1980], Angluin and Smith [1983], Osherson, Stob and Weinstein [1986], Wiehagen [1991], and Gasarch and Smith [1995], [1997] for surveys of many others, as well as for bibliographical references on them.

Historical background $\star$

Aristotle saw induction as one of the two ways of dialectic (together with deduction), defined it as a procedure leading from the particular to the universal (*Topics*, I, 12) and credited its discovery to Socrates (*Metaphysics*, XIII, 4).

The history of induction can be organized into three main chapters. The first deals with the dispute among **Stoics** and **Epicureans** reported in Philodemus' *De signis*. The object of the dispute was whether the legitimate form of reasoning was deduction or induction. One of the objections put forward by the Stoics was that *incomplete induction* (a generalization from a proper subset of the set of all possible data) can only draw uncertain conclusions, while *complete induction* (a generalization from the whole set of possible data)[6] is both theoretically reducible to deduction and practically impossible, if pursued by enumeration of the whole set of data.

Bacon [1620] agreed with the objection against complete induction and added that it is a dangerous practice, on the ground that a single mistake in the data might provoke the rejection of a correct generalization. However, Bacon thought that incomplete induction is not only possible, but can even reach certain conclusions, if pursued by means of an axiomatic analysis of the essence of the available data, through a classification by means of tables listing presence or absence of relevant features.

The second chapter of the history of induction in natural science took the form of a revival of the dispute among Stoics and Epicureans, whose roles were taken by **Rationalists** and **Empiricists**. In particular, in a letter to James Bernoulli (3.xii.1703), Leibniz reported the following *curve-fitting paradox*: since for any finite number of points there are always infinitely many curves going through them, any finite set of data is compatible with infinitely many inductive generalizations. The paradox raised the problem of which inductive generalization should be chosen, among the many available ones fitting the data, and it is has been twice reproposed in the 1950s, with much fanfare:

- First, by Wittgenstein [1953], in the following form: since any finite course of action is in accord with infinitely many rules, no universal rule can be learned by examples.
- Second, by Goodman [1954], in the following form: past observation on all emeralds so far discovered supports conclusions giving rise to opposite predictions, such as ‘all emeralds are green’ or ‘all emeralds are green if discovered so far, and blue if discovered from now on’.

Hume [1739] formulated a fundamental skeptic doubt: since necessity is something in the mind and not in the events, one can never demonstrate the necessity of a cause. This revived the problem of how to justify inductive generalizations, to which two different kinds of solutions were proposed:

- The first one, by Mill [1872], attempts to turn induction into a deduction, by adding principles about the world (such as ‘the future resembles the past’, or ‘space-time is homogeneous’) that would make inductive conclusions follow logically from their premises.
- The second one, by Keynes [1921], assigns to inductive generalizations, according to available evidence, probabilities that should converge to 1 as the generalizations are supported by more and more independent events.

[6]An example of complete induction in this sense is the ω-rule, that deduces $\forall x A(x)$ from the infinitely many exhaustive premises $A(0), A(1), \ldots$

Peirce distinguished between *abduction*, which is the creative formulation of generalizations, and *induction*, which is the process of validation of proposed generalizations (through tests and deductions). He also explicitly drew a conclusion that was implicit in previous work, namely that the possibility of self-correction (intrinsic in the combination of abduction and induction) allows success only in the limit.

The third chapter of the history of induction was written in the 1930s. In particular, the **Logical Positivists** of the Vienna Circle proposed the following *verification theory*, based on an idea of Wittgenstein: the meaning of a statement is in the conditions of its verification. In particular, a scientific hypothesis is meaningful only if it is verifiable. As a consequence, no generalization from a potentially infinite set of data (and hence, practically, no scientific law) is meaningful.

Popper [1934] replied with the following *falsification theory*: single observational events may prove hypotheses wrong, but no finite sequence of events can prove them correct. Thus induction is theoretically unjustifiable and becomes in practice the choice of the simplest generalization that resists falsification, on the ground that the simpler a hypothesis is (e.g. being of greater generality, or of lower probability), the easier it is to falsify it.

Carnap [1936] and [1937], partly accepting Popper's criticism, weakened the verification theory into the following *confirmation theory*: observational events provide, if not proofs, at least positive confirmations of scientific hypotheses. Thus induction becomes a choice of a generalization among the ones that confirm more evidence.

The theory of confirmation is subject to peculiar paradoxes, which appear when it is mixed with the usual logic of deduction:

- At the *propositional level*, Hesse [1974] noticed that the following apparently reasonable assumptions are contradictory:

  1. if $\alpha \to \beta$, then β confirms α
  2. if $\alpha \to \beta$ and γ confirms α, then γ confirms β.

  Indeed, since $(\alpha \wedge \beta) \to \alpha$, by 1 α confirms $\alpha \wedge \beta$. And since $(\alpha \wedge \beta) \to \beta$, by 2 α confirms β. But α and β are arbitrary, so that any proposition confirms any other.

- At the *predicate level*, Hempel [1945] noticed that the following assumption is paradoxical, if added to 2 above:

  3. $A(a) \wedge B(a)$ confirms $(\forall x)(A(x) \to B(x))$.

  Indeed, by 3 $\neg A(a) \wedge \neg B(a)$ confirms $(\forall x)(\neg B(x) \to \neg A(x))$. By 2, it then it also confirms $(\forall x)(A(x) \to B(x))$, which is equivalent to it (notice that only the following weak, and more convincing, form of 2 is needed: if γ confirms a proposition, it also confirms anything logically equivalent to it). Thus, for example, any white dove confirms that every raven is black.

Most of the mathematical work on inductive inference in the first half of the 20th Century, as well as much of that in the second, was based on probability (see Cohen [1980] for a survey). The shift that brings us to the subject of the present section was made by Solomonov [1964], [1964a]. He identified scientific theories to compact descriptions (in the sense of Kolmogorov's complexity, see Li and Vitanyi [1993] and Calude [1994]) of observations about the past and predictions about the future. By arithmetization, he then saw the process of inductive inference as the search of a program for a recursive function, based on finite initial segments of its values.

A little later, Gold [1967] started a study of language learnability from the perspective of Artificial Intelligence and saw the process of language learning as the search of a program for an r.e. set, based on finite initial segments of an enumeration of it.

Identification by next value

Our first notion formalizes the idea of a uniform method of prediction or extrapolation.

Definition VII.5.1 (Bardzin [1972], Blum and Blum [1975]) *A class $\mathcal{C}$ of total recursive functions is* **identifiable by next value** *($\mathcal{C} \in \boldsymbol{NV}$) if there is a total recursive function g (called a next-value function for $\mathcal{C}$) such that, for every $f \in \mathcal{C}$ and almost every n,*

$$f(n+1) = g(\langle f(0), \ldots, f(n) \rangle).$$

Notice that we allow a finite number of wrong predictions for each element of the class (i.e. g can take guesses and learn from its mistakes), in accord with the observation above that only limit processes are sufficiently general.

Theorem VII.5.2 Number-Theoretic Characterization of $\boldsymbol{NV}$ (Bardzin and Freivalds [1972]) *A class of total recursive function is in NV if and only if it is a subclass of an r.e. class of total recursive functions.*

Proof. A next-value function g allows the computation of a recursive function f, past the finitely many exceptions. Thus any function f for which g is a next-value function is of the following form, for some sequence number a (coding a list $\langle a_0, \ldots, a_n \rangle$ for some n):

$$f_a(x) = \begin{cases} a_x & \text{if } x \leq n \\ g(\langle f_a(0), \ldots, f_a(x-1) \rangle) & \text{otherwise.} \end{cases}$$

Any such f_a is recursive (by Course-of-Value Recursion) uniformly in a, and by the S^m_n-Theorem there is then a recursive function h such that $\varphi_{h(a)} = f_a$. Thus the class $\{f_a\}_{a\in\omega}$ is an r.e. class of total recursive functions. This shows that any class of recursive functions identifiable by next value is a subclass of an r.e. class of total recursive functions.

Conversely, for an r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$ of total recursive functions, we may suppose that it is closed under finite variants (since the closure of an r.e. class under finite variants is still r.e.). Let g be the recursive function defined as follows:

- on the empty list, g takes the value 0;
- on the list $\langle a_0, \ldots, a_n\rangle$, g takes the value $\varphi_{h(e)}(n+1)$, for the first e such that $\varphi_{h(e)}(x) = a_x$ for all $x \leq n$ (i.e. g takes the next value of the first function in the class that agrees with all values coded by the given list).

Since the class $\{\varphi_{h(e)}\}_{e\in\omega}$ is closed under finite variants, g is total. Since the class is r.e. (i.e. the functions $\varphi_{h(e)}$ are uniformly recursive), g is recursive. Moreover, g is a next-value function for every function in the given class (and hence in any subclass of it), by definition. $\square$

Exercise VII.5.3 *A class of total recursive functions is in NV if and only if it is a subclass of a class of total recursive functions r.e. in $\mathcal{K}$.* (Case and Smith [1983]) (Hint: if h is a function recursive in $\mathcal{K}$ then, by the Limit Lemma (IV.1.17), $h(e)$ is the limit of a recursive function $g(e,s)$. The class $\{\varphi_{h(e)}\}_{e\in\omega}$ is contained in the r.e. class $\{\varphi_{g(e,s)}\}_{e,s\in\omega}$, but some $\varphi_{g(e,s)}$ may be partial. One thus considers the r.e. class $\{\varphi_{f(e,s)}\}_{e,s\in\omega}$, where $\varphi_{f(e,s)}$ is the total function that on x takes the first convergent value in a dovetailed computation of $\varphi_{g(e,s)}(x)$, $\varphi_{g(e,s+1)}(x)$, ...)

Theorem VII.5.4 Complexity-Theoretic Characterization of *NV* (Adleman) *A class of total recursive functions is in NV if and only if it is a subclass of a complexity class (w.r.t. some complexity measure).*

Proof. By VII.5.2, any class in *NV* is contained in an r.e. class of total recursive functions $\mathcal{C} = \{\varphi_{h(e)}\}_{e\in\omega}$. Consider the associated set $\{\Phi_{h(e)}\}_{e\in\omega}$ of step-counting functions w.r.t. any measure, and notice that if

$$t(x) = \max_{e\leq x} \Phi_{h(e)}(x),$$

then t is recursive because h is recursive, and $\mathcal{C}$ is contained in the complexity class $\mathcal{C}_t$ named by t.

Conversely, given a recursive function t, notice that if φ_e is total and

$$(\forall_\infty x)[\Phi_e(x) \leq t(x)],$$

then there is a constant k such that

$$(\forall x)[\Phi_e(x) \leq t(x) + k].$$

Let g be the recursive function defined as follows:

- on the empty list, g takes the value 0
- on the list $\langle a_0, \ldots, a_n \rangle$, g takes the value $\varphi_e(n+1)$ for the smallest pair $\langle e, k \rangle$ defined as follows, if there is one, and 0 otherwise:
  - $\langle e, k \rangle \leq n$
  - for any $x \leq n+1$, $\Phi_e(x) \leq t(x) + k$
  - for any $x \leq n$, $\varphi_e(x) \simeq a_x$.

g is total recursive because all checks are recursive, and the second condition on $\langle e, k \rangle$ ensures that $\varphi_e(n+1)$ is defined.

It remains to show that g identifies by next value the complexity class $\mathcal{C}_t$. Suppose $f \in \mathcal{C}_t$: then there is a pair $\langle e, k \rangle$ such that, for all x,

$$\Phi_e(x) \leq t(x) + k \qquad \text{and} \qquad \varphi_e(x) \simeq f(x).$$

For any n greater than the minimal one of such pairs as well as, for each smaller pair $\langle e_1, k_1 \rangle$, of an argument x such that

$$\Phi_{e_1}(x) \leq t(x) + k_1 \;\Rightarrow\; \varphi_{e_1}(x) \not\simeq f(x),$$

g always outputs the correct value $f(n+1)$ on the list $\langle f(0), \ldots, f(n) \rangle$. $\square$

Exercise VII.5.5 *VII.5.4 is a corollary of VII.5.2.* (Hint: by VII.3.2, a complexity class is (not necessarily r.e., but) contained in an r.e. class of total recursive functions.)

VII.5.2 and VII.5.4, as well as their proofs, provide a paradigm for a number of other characterizations in the section. They also show that all the usual classes of total recursive functions dealt with in Chapter VIII are in NV, and that no class of total recursive functions containing functions with unbounded range, or functions arbitrarily difficult to compute (w.r.t. any complexity measure), is in NV.

The next observation shows that we need to look for other notions of inference.

Proposition VII.5.6 (Putnam [1963a]) *The class of all total recursive functions is not in NV.*

Proof. Given a recursive function g, let

$$f(0) = 0 \qquad \text{and} \qquad f(n+1) = g(\langle f(0), \ldots, f(n) \rangle) + 1.$$

Then f is recursive but g is not a next-value function for it, and hence for the class of all total recursive functions. $\square$

Exercises VII.5.7 a) *VII.5.6 is a corollary of VII.5.2.* (Hint: see II.2.1.)
b) *VII.5.6 is a corollary of VII.5.4.* (Hint: see VII.3.2.g.)

For more on the present topic, see Bardzin and Freivalds [1972], Podnieks [1974], [1975], Klette [1976], Werner [1975], Lindner and Werner [1976], Zeugmann [1983], [1983a], [1991].

Identification by consistent explanation

We now turn to notions that formalize the idea of a uniform method of explanation (via indices, that code descriptions of recursive functions in any of the formalisms discussed in Chapter I). In a first attempt we require that the explanations agree with the available data.

Definition VII.5.8 (Gold [1967]) *A class $\mathcal{C}$ of total recursive functions is* **identifiable by consistent explanation** *($\mathcal{C} \in \boldsymbol{EX_{cons}}$) if there is a total recursive function g (called a guessing function for $\mathcal{C}$) such that, for every sequence number $\langle a_0, \ldots, a_n \rangle$:*

- $\varphi_{g(\langle a_0, \ldots, a_n \rangle)}(x) \simeq a_x$ *for all $x \leq n$,*

and for every $f \in \mathcal{C}$:

- $\lim_{n \to \infty} g(\langle f(0), \ldots, f(n) \rangle)$ *exists, i.e.*

$$(\exists n_0)(\forall n \geq n_0)[g(\langle f(0), \ldots, f(n) \rangle) = g(\langle f(0), \ldots, f(n_0) \rangle)]$$

- $\varphi_{\lim_{n \to \infty} g(\langle f(0), \ldots, f(n) \rangle)} = f$.

In other words, $g(\langle f(0), \ldots, f(n) \rangle)$ provides a guess to an index of f consistent with the available information, and the guess stabilizes (from a certain point on) on an index of f.

The next result connects the two notions of identification introduced so far, and its two proofs suggest the two characterizations of EX_{cons} given in VII.5.14 and VII.5.16.

Proposition VII.5.9 (Gold [1967]) *$NV \subseteq EX_{cons}$.*

Proof. We can use the characterization of NV given in VII.5.2, and repeat almost verbatim the second half of its proof. Given an r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$ of total recursive functions, which we may suppose closed under finite variants, let g be the recursive function defined as follows:

- on the empty list, g takes the value 0;
- on the list $\langle a_0, \ldots, a_n \rangle$, g takes the value $h(e)$, for the first e such that $\varphi_{h(e)}(x) = a_x$ for all $x \leq n$ (i.e. g guesses the first function in the class that agrees with all values coded by the given list).

As in VII.5.2, g is total recursive. Moreover, g identifies by consistent explanation every function in the given class (and hence in any subclass of it).

Alternatively, we can use the characterization of NV given in VII.5.4, and again repeat almost verbatim the second half of its proof. Given a complexity class $\mathcal{C}_t$, let g be the recursive function defined as follows:

- on the empty list, g takes the value 0
- on the list $\langle a_0, \ldots, a_n \rangle$, g takes the value e for the smallest pair $\langle e, k \rangle$ defined as follows, if there is one, and 0 otherwise:
  - $\langle e, k \rangle \leq n$
  - for any $x \leq n$, $\Phi_e(x) \leq t(x) + k$ and $\varphi_e(x) \simeq a_x$.

As in VII.5.4, g is total recursive. Moreover, g identifies by consistent explanation every function in the given complexity class (and hence in any subclass of it). $\Box$

The difficulty in proving the opposite inclusion is the following: given a guessing function g for a class of functions $\mathcal{C}$, one might think of producing as a next-value function for $\mathcal{C}$ the one defined by

$$g_1(\langle a_0, \ldots, a_n \rangle) = \varphi_{g(\langle a_0, \ldots, a_n \rangle)}(n + 1),$$

i.e. to let the guessed program guess the next value. The problem with this is that $g(\langle a_0, \ldots, a_n \rangle)$ might be a program for a partial function (since we only know that the *final* guesses on initial segments of functions in $\mathcal{C}$ will be programs for total functions), and thus g_1 might not be total.

If one restricts the guesses to programs for total functions, then the problem disappears (see VII.5.10.a). One can argue that such a restriction is implicit in Popper's Refutability Principle (Popper [1934]), according to which incorrect scientific explanations should be refutable. Indeed, the unsolvability of the Halting Problem (II.2.7) makes it in general impossible to decide whether a partial function is undefined at a given argument, and thus to refute an explanation which is incorrect on such a basis. For more on this, see the discussion following VII.5.39.

Exercises VII.5.10 a) *If in Definition VII.5.8 of EX_{cons} the guessing function g is allowed to output only indices for total functions, then one obtains an alternative characterization of NV.* (Van Leeuwen, Bardzin) (Hint: the idea just sketched shows that if $\mathcal{C}$ is a class identifiable by consistent explanation via a function g that always outputs programs for total functions, then it is in NV. The converse holds by the proof of VII.5.9, once $g(\langle\ \rangle)$ is defined as any program for a total recursive function.)

b) *If in Definition VII.5.8 of EX_{cons} the guessed program is required only not to contradict the available data, as opposed to agree with them, then one obtains an alternative characterization of EX_{cons}.* (Fulk [1988]) (Hint: this amounts to replacing the condition

$$\varphi_{g(\langle a_0,\ldots,a_n\rangle)}(x) \simeq a_x$$

by

$$\varphi_{g(\langle a_0,\ldots,a_n\rangle)}(x)\downarrow\ \Rightarrow\ \varphi_{g(\langle a_0,\ldots,a_n\rangle)}(x) \simeq a_x,$$

for all $x \leq n$. One cannot simply patch up $\varphi_{g(\langle a_0,\ldots,a_n\rangle)}$ by assigning the values a_x to $x \leq n$, since this would produce a different program for every sequence $\langle a_0,\ldots,a_n\rangle$. The idea is to use the fact that if $g(\langle f(0),\ldots,f(n)\rangle)$ is the limit of g on f, then g is automatically consistent on any σ extending $\langle f(0),\ldots,f(n)\rangle$, because $\varphi_{g(\sigma)}$ is a total function, and in particular $\varphi_{g(\sigma)}(x) = \sigma(x)$ for any x in the domain of σ.

Given $\langle a_0,\ldots,a_n\rangle$, one first uses the Fixed-Point Theorem to obtain an e such that φ_e is defined as follows: if $x \leq n$, then $\varphi_e(x) \simeq a_x$; if $x > n$, then $\varphi_e(x) \simeq y$ if $\varphi_e(0),\ldots,\varphi_e(x-1)$ are all defined, and y is the smallest number such that

$$g(\langle \varphi_e(0),\ldots,\varphi_e(x-1)\rangle) = g(\langle \varphi_e(0),\ldots,\varphi_e(x-1),y\rangle).$$

Then one defines a consistent g_1 by letting $g_1(\langle a_0,\ldots,a_n\rangle)$ be $g_1(\langle a_0,\ldots,a_{n-1}\rangle)$, if a_n is the smallest y such that

$$g(\langle a_0,\ldots,a_{n-1}\rangle) = g(\langle a_0,\ldots,a_{n-1},y\rangle),$$

and e otherwise.)

We now show that if g is allowed to output indices for partial functions, then one is able to identify by consistent explanation more classes of functions. This proves that technique has stronger requirements than science, in the sense that to predict a class of phenomena one needs more than just an explanation of them.

Proposition VII.5.11 (Blum and Blum [1975]) $EX_{cons} - NV \neq \emptyset$.

Proof. We want to find a class $\mathcal{C}$ of total recursive functions which is in EX_{cons} but not in NV. Let

$$\mathcal{C} = \{\Phi_e : \Phi_e \text{ total}\},$$

i.e. the class of all total step-counting functions w.r.t. any measure.

To show $\mathcal{C} \in EX_{cons}$ we can note that $\mathcal{C}$ is contained in the class $\{\Phi_e\}_{e\in\omega}$, which is a measured set of partial recursive functions (see VII.2.9), and prove the following result, of independent interest: *any class of total recursive functions contained in a measured set of partial recursive functions is in EX_{cons}.*

The proof is a trivial extension of the first proof of VII.5.9. Given a measured set $\{\varphi_{h(e)}\}_{e\in\omega}$ of partial recursive functions, we may suppose that it is closed under finite variants (since a function that differs finitely from a function with recursive graph still has recursive graph). One can then define g as in VII.5.9, the only change being that on the list $\langle a_0, \ldots, a_n\rangle$ one looks for the first e such that, for all $x \leq n$, $\varphi_{h(e)}(x) \simeq a_x$ (as opposed to $\varphi_{h(e)}(x) = a_x$). Then g is recursive because, although $\varphi_{h(e)}(x)$ may be undefined, to check whether it is equal to a_x is a recursive procedure, by definition of measured set.

Alternatively, to show $\mathcal{C} \in EX_{cons}$ we can note that $\mathcal{C}$ is contained in the class $\{\Phi_e\}_{e\in\omega}$, which is an honest set of partial recursive functions (see VII.2.14.a), and prove the following result, of independent interest: *any class of total recursive functions contained in an honest set of partial recursive functions is in EX_{cons}.*

The proof is a trivial extension of the second proof of VII.5.9. Given the set of h-honest functions, one can define g as in VII.5.9, the only change being that on the list $\langle a_0, \ldots, a_n\rangle$ one looks for the smallest $\langle e, k\rangle \leq n$ such that, for all $x \leq n$,

$$\Phi_e(x) \leq h(x, a_x) + k \qquad \text{and} \qquad \varphi_e(x) \simeq a_x.$$

To show that $\mathcal{C} \notin NV$, we refer to VII.5.2. If $\mathcal{C}$ were a subclass of an r.e. class $\{\varphi_e\}_{e\in\omega}$ of total recursive functions, then we could recursively enumerate all total recursive functions by letting

$$\varphi_{g(e,i)}(x) = \begin{cases} \varphi_e(x) & \text{if } \Phi_e(x) \leq \varphi_{h(i)}(x) \\ 0 & \text{otherwise,} \end{cases}$$

thus contradicting II.2.1. Alternatively, we could obtain a recursive bound to the complexities of all total recursive functions by letting

$$t(x) = \max_{e \leq x} \varphi_{h(e)}(x),$$

thus contradicting VII.2.2. $\Box$

We turn now to a characterization of EX_{cons}. The characterization of NV given in VII.5.2 already used up all r.e. classes of total recursive functions, and thus we will look at r.e. classes of *partial* recursive functions.

Exercise VII.5.12 *If $\mathcal{C} \notin NV$ and $\mathcal{C}$ is a subset of an r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$ of partial recursive functions, then the problem of whether $\varphi_{h(e)}(x)\downarrow$ is undecidable.* (Wiehagen

[1991]) (Hint: otherwise, let h_1 be a recursive function such that

$$\varphi h_1(e)(x) = \begin{cases} \varphi_{h(e)}(x) & \text{if } \varphi_{h(e)}(x)\downarrow \\ 0 & \text{otherwise.} \end{cases}$$

Then $\{\varphi_{h_1(e)}\}_{e\in\omega}$ is an r.e. class of total recursive functions containing $\mathcal{C}$, and $\mathcal{C} \in NV$.)

We consider a notion that isolates just what is needed to make the proofs of VII.5.9 and VII.5.11 work.

Definition VII.5.13 *An r.e. class* $\{\varphi_{h(e)}\}_{e\in\omega}$ *is* **quasi-measured** *if there is a uniform recursive procedure to decide, given any* e *and any finite initial segment* σ*, whether* φ *extends* σ*.*

Obviously, a measured set is quasi-measured. To check whether $\varphi_{h(e)}$ extends σ, one simply checks whether $\varphi_{h(e)}(x) \simeq \sigma(x)$ for every x in the domain of σ.

But the converse does not hold. Quasi-measuredness allows us to check whether $\varphi_{h(e)}$ extends or not a given finite initial segment, but not whether it has a certain value on an isolated argument (unless we already know the values on all previous arguments).

Theorem VII.5.14 Number-Theoretic Characterization of EX_{cons} (Viviani) *A class of total recursive functions is in* EX_{cons} *if and only if it is a subclass of a quasi-measured set of partial recursive functions.*

Proof. The first proof of VII.5.11, showing that any class of total recursive functions contained in a measured set of partial recursive functions is in EX_{cons}, can easily be adapted to quasi-measured sets. First, we may suppose that the given quasi-measured set is closed under finite initial segments. Second, one can then define g as in VII.5.9, the only change being that on the list $\langle a_0, \ldots, a_n\rangle$ one looks for the first e such that $\varphi_{h(e)}$ extends $\langle a_0, \ldots, a_n\rangle$. Then g is recursive because, although we are not able to check the values of $\varphi_{h(e)}$ individually as in VII.5.11, we can recursively check whether $\varphi_{h(e)}$ agrees with the given initial segment.

Conversely, let $\mathcal{C}$ be identifiable by consistent explanation via g. Then every function f in $\mathcal{C}$ is of the following form, for some sequence number a (coding a list $\langle a_0, \ldots, a_n\rangle$ for some n):

$$f_a(x) \simeq \begin{cases} a_x & \text{if } x \le n \\ \varphi_{g(\langle a_0,\ldots,a_n\rangle)}(x) & \text{if } x > n \text{ and } g \text{ is not forced to change} \\ \text{undefined} & \text{otherwise} \end{cases}$$

(more precisely, $f \in \mathcal{C}$ is equal to f_a for any sequence number a coding the first n values of f, for any n such that $g(\langle f(0), \ldots, f(n)\rangle)$ has reached its limit). Any such f_a is partial recursive uniformly in a, and by the S^m_n-Theorem there is then a recursive function h such that $\varphi_{h(a)} = f_a$. Thus the class $\{f_a\}_{a\in\omega}$ is an r.e. class of partial recursive functions containing $\mathcal{C}$.

It remains to show that $\{f_a\}_{a\in\omega}$ is quasi-measured. Given a sequence number $a = \langle a_0, \ldots, a_n\rangle$ and an initial segment σ of length $m+1$, f_a extends σ if and only if:

1. either σ is contained in a (as a partial function), i.e. $\sigma(x) \simeq a_x$ for all $x \leq m$;

2. or σ extends a and $g(\langle a_0, \ldots, a_n\rangle)$ does not change on σ, i.e.

$$\begin{array}{rcl} g(\langle a_0, \ldots, a_n\rangle) & = & g(\langle a_0, \ldots, a_n, \sigma(n+1)\rangle) \\ & \cdots & \\ & = & g(\langle a_0, \ldots, a_n, \sigma(n+1), \ldots, \sigma(m)\rangle). \end{array}$$

The condition is obviously necessary, by definition of f_a, and we now show that it is also sufficient. In case 1, f_a agrees with a up to n, and hence with σ up to m. In case 2, f_a agrees with a, and hence with σ, up to n. For $n+1$, notice that

$$\varphi_{g(\langle a_0,\ldots,a_n\rangle)}(n+1) = \varphi_{g(\langle a_0,\ldots,a_n,\sigma(n+1)\rangle)}(n+1) = \sigma(n+1),$$

where the first equality holds because $g(\langle a_0, \ldots, a_n\rangle)$ does not change on σ, and the second does by consistency of g; this shows in particular that $f_a(n+1)$ is defined, because g is not forced to change, and that

$$f_a(n+1) = \sigma(n+1).$$

The proof for $n+2$ is similar, using the fact just proved that $f_a(n+1) = \sigma(n+1)$, and that g does not change on σ by hypothesis. In the same way one can proceed all the way to m. $\square$

As we have already done for the notion of measuredness in VII.5.13, we now consider a weakening of the notion of honesty. Recall from VII.2.13 that f is h-honest if

$$(\exists e)[f \simeq \varphi_e \ \wedge\ (\forall_\infty x)(\Phi_e(x) \leq h(x, \varphi_e(x))].$$

Definition VII.5.15 *If f and h are recursive functions, then f is* **quasi-*h*-honest** *if*

$$(\exists e)[f \simeq \varphi_e \ \wedge\ (\forall_\infty x)(\Phi_e(x) \leq h(x, \max_{y\leq x} \varphi_e(y)))].$$

Obviously, if f is h-honest and h is monotone, then f is quasi-h-honest because

$$\Phi_e(x) \leq h(x, \varphi_e(x)) \leq h(x, \max_{y \leq x} \varphi_e(y)).$$

But the converse does not hold. Quasi-h-honesty provides a bound to $\Phi_e(x)$ in terms only of $\max_{y \leq x} \varphi_e(y)$, not just of $\varphi_e(x)$.

Theorem VII.5.16 Complexity-Theoretic Characterization of $\boldsymbol{EX_{cons}}$ *A class of total recursive functions is in EX_{cons} if and only if it is a class of quasi-h-honest functions, for some recursive function h.*

Proof. The second proof of VII.5.11, showing that any class of total recursive functions contained in a set of honest functions is in EX_{cons}, can easily be adapted to quasi-honest sets, by substituting $h(x, \max_{y \leq x} a_y)$ for $h(x, a_x)$.

Conversely, let $\mathcal{C}$ be identifiable by consistent explanations via g. Then one can define the following function:

$$h(x, z) = \max\{\Phi_{g(\langle a_0, \ldots, a_x \rangle)}(x) : z = \max_{y \leq x} a_y\}.$$

Since g is consistent, $\varphi_{g(\langle a_0, \ldots, a_x \rangle)}(x) \simeq a_x$, and so $\Phi_{g(\langle a_0, \ldots, a_x \rangle)}(x)$ is defined. Moreover, there are only finitely many sequence numbers $\langle a_0, \ldots, a_x \rangle$ such that $z = \max_{y \leq x} a_y$, and hence h is total recursive.

If $f \in \mathcal{C}$, then g has a limit e on f, and $\varphi_e \simeq f$, because g identifies $\mathcal{C}$. If x_0 is a point after which g does not change anymore on f, then

$$\Phi_e(x) \leq h(x, \max_{y \leq x} f(y))$$

for all $x \geq x_0$. Thus e is a witness of the fact that f is quasi-h-honest. □

The next result is a corollary of any of the previous ones (see VII.5.18), but a direct proof is simple and instructive.

Proposition VII.5.17 (Gold [1967]) *The class of all total recursive functions is not in EX_{cons}.*

Proof. Let g be a recursive function such that

$$f \text{ total recursive } \Rightarrow \varphi_{\lim_{n \to \infty} g(\langle f(0), \ldots, f(n) \rangle)} = f.$$

We define a total recursive function f such that $\lim_{n \to \infty} g(\langle f(0), \ldots, f(n) \rangle)$ does not exist, contradicting the hypothesis on g. The idea is that f will repeat the same value a number of times sufficient to force a change in g. Since we do this infinitely often, then g changes infinitely often on f, and thus it has no limit.

Start with $f(0) = 0$. The functions f_0 identically equal to 0, and f_1 equal to 0 for $x = 0$ and equal to 1 afterwards, are total recursive and different. By hypothesis on g,

$$\lim_{n\to\infty} g(\langle f_0(0), \ldots, f_0(n)\rangle) \neq \lim_{n\to\infty} g(\langle f_1(0), \ldots, f_1(n)\rangle).$$

Thus there must exist n and i ($i = 0$ or $i = 1$) such that

$$g(\langle f(0)\rangle) \neq g(\langle f_i(0), \ldots, f_i(n)\rangle),$$

and we can effectively find them. Let $f(x)$ be equal to $f_i(x)$ for every $x \leq n$, and iterate the procedure (by extending, at each step, the values already obtained by a sequence of either 0's or 1's). $\square$

Exercises VII.5.18 a) *VII.5.17 is a corollary of VII.5.14.* (Hint: given a quasi-measured set $\{\varphi_{h(e)}\}_{e\in\omega}$, define f total recursive not in it as follows. Having the values $f(0), \ldots, f(x-1)$, see if $\varphi_{h(x)}$ extends $\langle f(0), \ldots, f(x-1), 0\rangle$. If so, then let $f(x) = 1$. If not, let $f(x) = 0$. Then f differs from $\varphi_{h(x)}$ on an argument $\leq x$.)

b) *VII.5.17 is a corollary of VII.5.16.* (Hint: by VII.2.3, since the function $t(x) = h(x, 0) + h(x, 1)$ is a bound to some complexity of every 0,1-valued quasi-h-honest function.)

For more on the present notion, see Wiehagen and Liepe [1976], Wiehagen [1976], [1978], [1991], Kugel [1977], Freivalds and Wiehagen [1979], Angluin [1980], Jantke and Beik [1981], Zeugmann [1983], [1983a], [1991], Fulk [1988], Jantke [1991].

Identification by reliable explanation

The next notion formalizes the idea of a method of explanation that never permanently settles on a false hypothesis, and thus gives indirect information about its mistakes.

Definition VII.5.19 (Blum and Blum [1975]) *A class $\mathcal{C}$ of total recursive functions is* **identifiable by reliable explanation** *($\mathcal{C} \in \boldsymbol{EX_{rel}}$) if there is a total recursive function g such that, for every $f \in \mathcal{C}$:*

- $\lim_{n\to\infty} g(\langle f(0), \ldots, f(n)\rangle)$ *exists, i.e.*

$$(\exists n_0)(\forall n \geq n_0)[g(\langle f(0), \ldots, f(n)\rangle) = g(\langle f(0), \ldots, f(n_0)\rangle)]$$

and for every total recursive f:[7]

[7] This is the most we can ask in our setting, since $g(\langle f(0), \ldots, f(n)\rangle)$ can be defined for all n only if f is total, and its limit can be an index of f only if f is recursive.

- $\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)$ *exists* $\Rightarrow$ $\varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)} = f$.

In other words, $g(\langle f(0),\ldots,f(n)\rangle)$ *provides a guess to an index of* f*, which stabilizes (from a certain point on) on an index of* f *whenever it stabilizes, and it does stabilize if* $f \in \mathcal{C}$*.*

Notice that, despite the fact that we do not require from our guesses that they be consistent with the available information, such information cannot be disregarded, lest we proceed independently of f and never be able to stabilize our guess.

Proposition VII.5.20 $EX_{cons} \subseteq EX_{rel}$.

Proof. Let g identify $\mathcal{C}$ by consistent explanation. It is enough to show that

$$\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle) \text{ exists } \Rightarrow \varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)} = f.$$

If $\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)$ exists, let n_0 be such that

$$(\forall n \geq n_0)[g(\langle f(0),\ldots,f(n)\rangle) = g(\langle f(0),\ldots,f(n_0)\rangle).$$

For every x, if n is greater than both n_0 and x, then

$$\varphi_{g(\langle f(0),\ldots,f(n_0)}(x) \simeq \varphi_{g(\langle f(0),\ldots,f(n)\rangle)}(x) \simeq f(x),$$

where the first equality holds because $n \geq n_0$, and the second one because $n \geq x$ and g is consistent. $\square$

Having showed that reliable identification is at least as powerful as consistent identification, we now show that it is strictly more powerful.

Proposition VII.5.21 (Blum and Blum [1975], Fulk [1988]) $EX_{rel} - EX_{cons} \neq \emptyset$.

Proof. We want to find a class $\mathcal{C}$ of total recursive functions which is in EX_{rel} but not in EX_{cons}. Let

$$\mathcal{C} = \{f : (\exists e)[f \simeq \varphi_e \ \wedge \ (\forall x)(\Phi_e(x) \leq f(x+1))]\}.$$

To show that $\mathcal{C} \in EX_{rel}$, define g as follows: $g(\langle\ \rangle) = 0$; given $\langle a_0,\ldots,a_n\rangle$, look for the smallest $e \leq n$ such that $\Phi_e(x) \leq a_{x+1}$ and $\varphi_e(x) \simeq a_x$ for all $x < n$, and let $g(\langle a_0,\ldots,a_n\rangle)$ be such an e if one exists, and n otherwise.[8]

[8]Notice that this does not even imply that $\varphi_e(n)$ is defined, let alone that it is equal to a_n. Thus the proposed identification procedure is not necessarily consistent (by the second part of the proof, it is provably not consistent).

The function g is recursive. Indeed, to compute it first we check recursively whether $\Phi_e(x) \leq a_{x+1}$. If so, then we compute $\varphi_e(x)$, which is defined because $\Phi_e(x)$ is finite, and compare it to a_x. Since this is done only for $e \leq n$ and $x < n$, $g(\langle a_0, \ldots, a_n \rangle)$ can be effectively computed.

If $f \in \mathcal{C}$, then the limit of $g(\langle f(0), \ldots, f(n) \rangle$ exists, and it actually is the smallest e witnessing membership of f in $\mathcal{C}$.

Finally, if $\lim_{n\to\infty} g(\langle f(0), \ldots, f(n) \rangle)$ exists then, by the second case in the definition of g, it must be because the first case is applied for all sufficiently large n. By definition, for all such n and $x < n$,

$$\varphi_{g(\langle f(0),\ldots,f(n)\rangle)}(x) \simeq f(x).$$

Thus, for all x,

$$\varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)}(x) = f(x).$$

To show that $\mathcal{C} \notin EX_{cons}$, suppose g is a consistent recursive guessing function such that

$$f \in \mathcal{C} \;\Rightarrow\; f = \varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)}.$$

The idea is to construct $f \in \mathcal{C}$ such that either $\lim_{n\to\infty} g(\langle f(0), \ldots, f(n) \rangle)$ does not exist (because g changes its guess infinitely often), or it exists but it is not an index of f (because f differs from it on some argument).

By the S_n^m-Theorem, there is a recursive function t such that $\varphi_{t(e)}$ is defined as follows. If $x = 0$, then $\varphi_{t(e)}(0) \simeq 0$. If $x > 0$, wait until all of $\varphi_e(0), \ldots, \varphi_e(x-1)$, and hence also $\Phi_e(x-1)$, are defined. There are two possible cases:

1. If $g(\langle \varphi_e(0), \ldots, \varphi_e(x-1) \rangle) \neq g(\langle \varphi_e(0), \ldots, \varphi_e(x-1), \Phi_e(x-1)+1 \rangle)$, then we define $\varphi_{t(e)}(x) \simeq \Phi_e(x-1)+1$, thus ensuring that g has changed its value once.

2. Otherwise, we define $\varphi_{t(e)}(x) \simeq \Phi_e(x-1)$, thus ensuring that $\varphi_{t(e)}$ is different from the function guessed by g on the initial segment of $\varphi_{t(e)}$ of length x.

By the Fixed-Point Theorem, there is e such that $\varphi_e \simeq \varphi_{t(e)}$. Let $f \simeq \varphi_e$. Then f is total by induction, and in $\mathcal{C}$ because $\Phi_e(x) \leq f(x+1)$ for every x. There are now two possible cases:

- $\lim_{n\to\infty} g(\langle f(0), \ldots, f(n) \rangle)$ *does not exist*
  Then g does not identify f by consistent explanation.

- $\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)$ *exists*
  If $g(\langle f(0),\ldots,f(x)\rangle)$ is this limit, then case 1 cannot take place at x, otherwise g would have changed its value at least once more, and so
  $$\varphi_{g(\langle f(0),\ldots,f(x-1)\rangle)}(x) \simeq \varphi_{g(\langle f(0),\ldots,f(x-1),\Phi_e(x-1)+1\rangle)}(x) \simeq \Phi_e(x-1)+1,$$
  where the second equality holds by consistency of g. But since case 1 does not take place, $f(x) = \Phi_e(x-1)$; thus f differs from $\varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)}$ at x. $\square$

We do not know of any number-theoretic characterization of EX_{rel}. For a complexity-theoretic characterization, a hint comes from the notion of quasi-honesty used in VII.5.16, and the observation that the expression
$$h(x, \max_{y\leq x} \alpha(x)),$$
used in it defines a partial recursive functional $H(\alpha, x)$.

Definition VII.5.22 *If f is a recursive function and H is a partial recursive functional, then f is **H-honest** if*
$$(\exists e)[f \simeq \varphi_e \ \wedge\ (\forall_\infty x)(H(\varphi_e,x)\downarrow \wedge\, \Phi_e(x) \leq H(\varphi_e,x))].$$

Theorem VII.5.23 Complexity-Theoretic Characterization of EX_{rel} (Blum and Blum [1975]) *A class $\mathcal{C}$ of total recursive functions is in EX_{rel} if and only if it is a class of H-honest functions, for some partial recursive functional H which is total on all total recursive functions.*

Proof. Given a partial recursive functional H total on all total recursive functions, notice that if φ_e is total and H-honest, then there is a constant k such that
$$(\forall x)[\Phi_e(x) \leq H(\varphi_e,x)+k].$$
One can define a recursive function g that reliably identifies the class of all total H-honest functions, as follows:

- on the empty list, g takes the value 0
- on the list $\langle a_0,\ldots,a_n\rangle$, g takes the value e for the smallest pair $\langle e,k\rangle$ defined as follows, if there is one, and n otherwise:
  - $\langle e,k\rangle \leq n$
  - for all $x \leq n$, if $H(\langle a_0,\ldots,a_n\rangle, x)$ converges in at most n steps (where $\langle a_0,\ldots,a_n\rangle$ is considered as the partial function giving value a_x to $x \leq n$, and undefined otherwise), then
  $$\Phi_e(x) \leq H(\langle a_0,\ldots,a_n\rangle, x)+k \qquad \text{and} \qquad \varphi_e(x) \simeq a_x.$$

g is total recursive because all checks are recursive. It remains to show that g identifies by reliable explanation the class of all total H-honest functions.

First, suppose f is recursive, $\lim_{n\to\infty} g(\langle f(0),\dots,f(n)\rangle)$ exists, and e is its value. Since f is total, for every x there is $n \geq x$ such that $H(\langle f(0),\dots,f(n)\rangle, x)$ converges in at most n steps;[9] after the stage in which g has reached its final value e, the construction ensures that $\varphi_e(x) \simeq f(x)$. Since this holds for every x, $\varphi_e \simeq f$.

Second, suppose f is a total H-honest function. Then there is a pair $\langle e, k\rangle$ such that, for all x,

$$\Phi_e(x) \leq H(f,x) + k \qquad \text{and} \qquad \varphi_e(x) \simeq f(x),$$

and hence g must have a limit on f, since it must stop changing guesses when it hits such a pair $\langle e, k\rangle$, if not before.

In the opposite direction, let $\mathcal{C}$ be identifiable by reliable explanation via g, and define

$$H(\alpha, x) \simeq \begin{cases} \Phi_{g(\langle\alpha(0),\dots,\alpha(x)\rangle)}(x) & \text{if } \varphi_{g(\langle\alpha(0),\dots,\alpha(x)\rangle)}(x) \text{ converges first} \\ 0 & \text{if } g \text{ changes its mind on } \alpha \text{ after } x \text{ first} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

H is a partial recursive functional by definition.

If f is a total recursive function, we show that $H(f,x)$ is defined for any x, so that H is total on the total recursive functions. Either $g(\langle f(0),\dots,f(x)\rangle)$ changes after x, and then $H(f,x)$ is defined by the second clause if not otherwise. Or $g(\langle f(0),\dots,f(x)\rangle)$ is the limit of g on f and hence, by reliability, an index of the total function f, so that $H(f,x)$ is defined by the first clause.

Finally, if $f \in \mathcal{C}$, then g has a limit e on f, and $\varphi_e \simeq f$, because it identifies $\mathcal{C}$. If x_0 is a point after which g does not change anymore on f, then $H(f,x) \simeq \Phi_e(x)$ for all $x \geq x_0$, because $\varphi_e(x)$ converges (since f is total). Thus e is a witness of the fact that f is H- honest. $\square$

The proof of VII.5.17 does not use any reliability hypothesis, and it thus shows that *the class of all total recursive functions is not in* EX_{rel}.

Exercise VII.5.24 *The last observation is a corollary of VII.5.23.* (Hint: using the fact that H is total on all total recursive functions, build a recursive increasing sequence $\{x_n\}_{n\in\omega}$ with the property that, for every n, $H(x_n, f)$ converges for any function f equal to 0 on every $x < x_{n+1}$ such that $x \neq x_i$ for all $i \leq n$, and equal

[9]More precisely, there is n_1 such that $H(\langle f(0),\dots,f(n_1)\rangle, x)$ converges, and hence there is n_2 such that $H(\langle f(0),\dots,f(n_1)\rangle, x)$ converges in at most n_2 steps. Then, by monotonicity of recursive functionals (II.3.13), $H(\langle f(0),\dots,f(n)\rangle, x)$ converges in at most n steps, for $n = \max\{n_1, n_2, x\}$.

to 0 or 1 on any such x_i. Let $t(x_n)$ be the maximum of the values $H(x_n, f)$ for such functions f. Using the methods of Rabin's Theorem VII.2.3, build a 0,1-valued recursive function f such that if $(\forall_\infty x)(\Phi_e(x) \leq t(x))$, then $f \not\simeq \varphi_e$, by diagonalizing on the x_n. If $\varphi_e \simeq f$, then $\Phi_e(x) > t(x)$ for infinitely many x and f is not H-honest.)

For more on the present notion, see Blum and Blum [1975], Minicozzi [1976], Zeugmann [1983], [1983a], [1991], Kinber and Zeugmann [1985], Fulk [1988].

Identification by explanation

There is an obvious way of relaxing the definitions of EX_{cons} and EX_{rel}, which consists in just dropping any consistency or reliability requirement.

Definition VII.5.25 (Gold [1967]) *A class $\mathcal{C}$ of total recursive functions is* **identifiable by explanation** *($\mathcal{C} \in \boldsymbol{EX}$) if there is a total recursive function g such that, for every $f \in \mathcal{C}$:*

- $\lim_{n\to\infty} g(\langle f(0), \ldots, f(n)\rangle)$ *exists, i.e.*

$$(\exists n_0)(\forall n \geq n_0)[g(\langle f(0), \ldots, f(n)\rangle) = g(\langle f(0), \ldots, f(n_0)\rangle)]$$

- $\varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)} = f$.

In other words, $g(\langle f(0), \ldots, f(n)\rangle)$ provides a guess to an index of f, and the guess stabilizes (from a certain point on) on an index of f.

Exercises VII.5.26 Best guessing functions. A guessing function g is **best for $\boldsymbol{\mathcal{C}}$** if it identifies $\mathcal{C}$ by explanation, and there is no other guessing function g_1 that identifies $\mathcal{C}$ and such that:

- for all $f \in \mathcal{C}$, g_1 does not converge on f after g does
- for some $f \in \mathcal{C}$, g_1 converges on f before g does.

a) *There are classes $\mathcal{C}$ admitting best guessing functions.* (Gold [1967]) (Hint: any r.e. class of total recursive functions, see VII.5.9.)

b) *A guessing function g is best for $\mathcal{C}$ if and only if it is consistent, it always outputs indices for functions in $\mathcal{C}$ and it never changes a correct guess.* (Gold [1967], Jantke and Beik [1981]) (Hint: suppose g satisfies the properties and g_1 is another guessing function for $\mathcal{C}$. If g_1 converges on some $f \in \mathcal{C}$ before g does, we want to show that g converges on some $f_1 \in \mathcal{C}$ before g_1 does. Let n and n_1 ($n_1 < n$) be the first places where g and g_1 converge on f. By the third property of the statement, n is the first place where g produces a right index for f. Thus, g must produce at n_1 an index for another function f_1, which is in $\mathcal{C}$ by the second property and agrees with f up to n_1 by the first property. Since g_1 produces an index for f at n_1, it must converge on f_1 at some later stage. Thus g converges on f_1 before g_1 does.)

Proposition VII.5.27 (Bardzin [1974], Blum and Blum [1975]) *$EX_{rel} \subset EX$.*

Proof. The inclusion is obvious by definition. To show that it is proper, we want to find a class $\mathcal{C}$ of total recursive functions which is in EX but not in EX_{cons}. Let

$$\mathcal{C} = \{f : f = \varphi_{f(0)}\},$$

i.e. the class of total recursive functions such that $f(0)$ is an index of f.

$\mathcal{C} \in EX$, by letting $g(\langle\ \rangle) = 0$ and $g(\langle a_0, \ldots, a_n\rangle) = a_0$. In other words, every function in $\mathcal{C}$ gives away a program for itself as its first value, thus making identification by explanation trivial.

To show that $\mathcal{C} \notin EX_{rel}$, suppose g is a reliable recursive guessing function such that

$$f = \varphi_{f(0)} \ \Rightarrow\ f = \varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)}.$$

By reliability, for any f

$$\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle) \text{ exists } \Rightarrow\ f = \varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)}.$$

Thus we cannot construct an $f \in \mathcal{C}$ such that if $\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)$ exists, then it is not an index of f (as was one option in the proof of VII.5.21), and we are forced to find a function f such that

$$f = \varphi_{f(0)} \qquad \text{and} \qquad \lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle) \text{ does not exist.}$$

For this we proceed in a way similar to the proof of VII.5.17.

By the S^m_n-Theorem, there is a recursive function t such that $\varphi_{t(e)}$ is defined as follows. Start with $\varphi_{t(e)}(0) \simeq e$. Consider the functions f_i ($i = 0, 1$) equal to e for $x = 0$, and identically equal to i for $x > 0$. Of the two limits

$$\lim_{n\to\infty} g(\langle f_1(0),\ldots,f_1(n)\rangle) \qquad \text{and} \qquad \lim_{n\to\infty} g(\langle f_2(0),\ldots,f_2(n)\rangle)$$

either at least one does not exist, or they both exist and are different (by reliability of g, since the two functions f_0 and f_1 are different). Thus there must exist n and i ($i = 0$ or $i = 1$) such that

$$g(\langle \varphi_{t(e)}(0)\rangle) \neq g(\langle f_i(0),\ldots,f_i(n)\rangle),$$

and we can effectively find them. Let $\varphi_{t(e)}(x)$ be equal to $f_i(x)$ for every $x \leq n$, and iterate the procedure (by extending, at each step, the values already obtained by a sequence of either 0's or 1's).

By the Fixed-Point Theorem, there is e such that $\varphi_e \simeq \varphi_{t(e)}$. Let $f \simeq \varphi_e$. Then f is in $\mathcal{C}$ because its value on 0 is an index for it, and g does not identify f because, by construction, $\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)$ does not exist. $\square$

Notice that the class $\{f : f = \varphi_{f(0)}\}$ used in the previous proof is a subclass of the r.e. class of partial recursive functions $\{\varphi_{h(e)}\}_{e\in\omega}$ defined as follows:

$$\varphi_{h(e)}(x) \simeq \begin{cases} e & \text{if } x = 0 \\ \varphi_e(x) & \text{otherwise.} \end{cases}$$

We can effectively tell apart all pairs of members of such a class, since they all differ on their first arguments.

We turn now to a characterization of EX, in terms of the following generalization of the property just noticed.

Definition VII.5.28 *An r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$ is* **effectively separable** *if there is a uniform recursive procedure to determine, for every different e and i, an upper bound to an argument on which $\varphi_{h(e)}$ and $\varphi_{h(i)}$ disagree.*

The definition implies that an effectively separable class $\{\varphi_{h(e)}\}_{e\in\omega}$ is enumerated without repetitions by h.

Theorem VII.5.29 First Number-Theoretic Characterization of *EX* (Wiehagen and Jung [1977], Wiehagen [1978]) *A class $\mathcal{C}$ of total recursive functions is in EX if and only if it is a subclass of an effectively separable r.e. class of partial recursive functions.*

Proof. Given an effectively separable r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$, there is a recursive function d (for 'disagreement') such that, for every different e and i, there is $x \leq d(e,i)$ such that $\varphi_{h(e)}(x) \not\simeq \varphi_{h(i)}(x)$. One can define a recursive function g that identifies by explanation any subclass $\mathcal{C}$ of the given class, as follows:

- on the empty list, g takes the value $h(0)$
- suppose the value of g on the list $\langle a_0, \ldots, a_{n-1}\rangle$ was $h(e)$; then on the list $\langle a_0, \ldots, a_n\rangle$ the value of g is still $h(e)$, unless there is an i as follows, in which case g takes the value $h(e+1)$:
  - i is different from e
  - $i \leq n$
  - $d(e,i) \leq n$
  - for all $x \leq d(e,i)$, $\varphi_{h(i)}(x) \simeq a_x$, with all computations convergent in at most n steps.

The function g is total recursive because all checks are recursive.

It remains to show that $\mathcal{C}$ is identified by explanation via g. Let $f \in \mathcal{C}$ be given. If $g(\langle f(0), \ldots, f(m)\rangle) = h(e)$ is not an index of f, then there is i

such that $\varphi_{h(i)} = f$, because $\mathcal{C} \subseteq \{\varphi_{h(e)}\}_{e\in\omega}$. Indeed, for a sufficiently big n the conditions above will be satisfied, and g will switch to the value $h(e+1)$ because the fact that, up to $d(e,i)$, $\varphi_{h(i)}$ agrees with f while $\varphi_{h(e)}$ disagrees with $\varphi_{h(i)}$ is enough to tell us that $h(e)$ was not a correct guess. As soon as g hits an i such that $\varphi_{h(i)} = f$, g will no longer change its guesses, and this must happen because one successively tries all $h(e)$'s.[10]

In the opposite direction, let $\mathcal{C}$ be identifiable by explanation via g. Then every function in $\mathcal{C}$ is of the following form, for some sequence number a (coding a list $\langle a_0, \dots, a_n\rangle$ for some n):

$$f_a(x) \simeq \begin{cases} a_x & \text{if } x \le n \\ \varphi_{g(\langle a_0,\dots,a_n\rangle)}(x) & \text{otherwise} \end{cases}$$

(more precisely, $f \in \mathcal{C}$ is equal to f_a for any sequence number a coding the first n values of f, for any n such that $g(\langle f(0), \dots, f(n)\rangle)$ has reached its limit). Any such f_a is partial recursive uniformly in a, and by the S^m_n-Theorem there is then a recursive function h such that $\varphi_{h(a)} = f_a$. Thus the class $\{f_a\}_{a\in\omega}$ is an r.e. class of partial recursive functions containing $\mathcal{C}$. But not all members of it are different (since, by the parenthetical remark above, each function in $\mathcal{C}$ appears infinitely often in the class).

Incompatible sequence numbers are not problematic, since the associated functions agree with them, and hence they differ among each other. Among the compatible sequence numbers, we can certainly cut down a number of repetitions, by considering only those for which g provides new guesses. They form an r.e. class, and hence so does the set of associated functions. But we can still have two compatible sequence numbers a and b such that $f_a = f_b$. For example, both $g(a)$ and $g(b)$ might be the final guess of an index of a function in $\mathcal{C}$, but in between them g could have changed its mind.

We thus take two complementary actions. On the one hand, we start defining a new function only when we hit a sequence number that provides a new guess of g. On the other hand, we stop defining the new function as soon as we discover that the guess of g changes. In other words, for any sequence number $a = \langle a_0, \dots, a_n\rangle$ such that

$$g(\langle a_0, \dots, a_{n-1}\rangle) \neq g(\langle a_0, \dots, a_n\rangle)$$

we let:

$$f_a(x) \simeq \begin{cases} a_x & \text{if } x \le n \\ \varphi_{g(\langle a_0,\dots,a_n\rangle)}(x) & \text{if } x > n \text{ and } g \text{ is not forced to change} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

[10]Notice that if one chose instead $h(i)$ as the new guess in the definition of g, there would be no guarantee that the procedure would converge to an index of f, because every time one could hit an i such that $\varphi_{h(i)}$ is not f, but looks like it up to $d(e,i)$, in particular sufficiently so to be able to rule out $\varphi_{h(e)}$ as a possible guess.

As argued above, we thus obtain an r.e. class of partial recursive functions containing $\mathcal{C}$. It only remains to show that there is a recursive function d such that, if a and b are sequence numbers among the ones considered, then $d(a,b)$ is an upper bound to an argument on which f_a and f_b disagree.

It is enough to let $d(a,b)$ be the maximum of the lengths of a and b. There are two cases to consider:

- If a and b are incompatible, then they differ on some component below $d(a,b)$. But f_a and f_b agree with a and b, respectively, and so they differ on some argument below $d(a,b)$.

- If a and b are compatible, suppose e.g. that a is contained in b, and thus $d(a,b)$ is just the length of b.

  Since both a and b have been considered, g must have changed its guess on some intermediate sequence number, i.e. on some initial segment of b of length $n < d(a,b)$. Then either f_a did not agree with b up to n, and then it differs from f_b for one reason (because by definition f_b does agree with b up to its length), or it did agree, and then it differs from f_b for another reason (because by definition f_a stops being defined at n if not before, while f_b is defined there). $\square$

We now introduce a variation of VII.5.28.

Definition VII.5.30 *An r.e. class $\{\varphi_e\}_{e\in\omega}$ of partial recursive functions is* **effectively discrete** *if there is a uniform procedure to determine, for every e and all almost all i, an upper bound to arguments on which $\varphi_{h(e)}$ disagrees with $\varphi_{h(i)}$.*

Notice how effective discreteness relates to effective separability. It is weaker, because it does not require the existence of a recursive enumeration of the given class without repetitions, and it allows instead for finitely many repetitions of each functions. And it is stronger, because it provides a bound that depends only on e and not on i. The next result shows that weakening and strengthening compensate, so that the new notion still characterizes the same class as the old one.

Theorem VII.5.31 Second Number-Theoretic Characterization of *EX* (Freivalds, Kinber and Wiehagen [1984]) *A class $\mathcal{C}$ of total recursive functions is in EX if and only if it is a subclass of an effectively discrete r.e. class of partial recursive functions.*

Proof. Given an effectively discrete r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$, there is a recursive function d (for 'discreteness') such that, for every e, there are at most finitely many i such that $\varphi_{h(e)}(x) \simeq \varphi_{h(i)}(x)$ for all $x \leq d(e)$. One can define a recursive function g that identifies by explanation any subclass $\mathcal{C}$ of the given class, as follows:

- on the empty list, g takes the value 0
- on the list $\langle a_0, \ldots, a_n \rangle$, g takes the value 0, unless there are indices e such that:
  - $e \leq n$
  - $d(e) \leq n$
  - for all $x \leq d(e)$, $\varphi_{h(e)}(x) \simeq a_x$ in at most n steps
  - for all x such that $d(e) < x \leq n$, if $\varphi_{h(e)}(x)$ converges in at most n steps, then $\varphi_{h(e)}(x) \simeq a_x$.

  In this case one lets $I_{\langle a_0,\ldots,a_n \rangle}$ be the set of the $h(e)$'s corresponding to all such e's and defines $g(\langle a_0, \ldots, a_n \rangle)$ as an index of the function that, on any input x, dovetails computations of $\varphi_{h(e)}(x)$ for all $h(e) \in I_{\langle a_0,\ldots,a_n \rangle}$ and outputs the first convergent value.

The function g is total recursive because all checks are recursive.

It remains to show that $\mathcal{C}$ is identified by explanation via g. Given any $f \in \mathcal{C}$, the definition of g and the fact that $\{\varphi_{h(e)}\}_{e \in \omega}$ is effectively discrete imply that, for any sufficiently large n, the sets $I_{\langle f(0),\ldots,f(n) \rangle}$ reach a finite limit, containing at least one index of f and only indices of functions compatible with it (because if $\varphi_{h(e)} = f$ and $\varphi_{h(i)}$ agrees with $\varphi_{h(e)}$ up to $d(e)$, then either $\varphi_{h(i)} \subseteq f$ or, for some x, $\varphi_{h(i)}(x)$ eventually converges to a value different from $f(x)$). From that point on, $g(\langle f(0), \ldots, f(n) \rangle)$ will always be the same index of f.

In the opposite direction, let $\mathcal{C}$ be identifiable by explanation via g. By letting $d(a)$ be the length of a, one sees that the r.e. class $\{f_a\}_{a \in \omega}$ defined in the proof of VII.5.29 is effectively discrete. $\Box$

Exercise VII.5.32 *A class $\mathcal{C}$ of total recursive functions is in EX if and only if there is a partial recursive functional F such that, for all $f \in \mathcal{C}$, the limit of $F(f, n)$ converges to an index of f.* (Gold [1967]) (Hint: given g, let $F(\alpha, n)$ be $g(\langle \alpha(0), \ldots, \alpha(n) \rangle)$. Conversely, given F, let $g(\langle a_0, \ldots, a_n \rangle)$ be $F(\langle a_0, \ldots, a_n \rangle, n)$, where $\langle a_0, \ldots, a_n \rangle$ is considered as a partial function, if it converges in at most n steps, and 0 otherwise.)

Turning now to complexity-theoretic characterizations, the most natural weakening of VII.5.23 is obtained by dropping the restriction that the functional H be total on all total recursive functions. This is however too weak, and will be used in VII.5.42 to characterize EX^*. For a characterization of EX, the following stronger notion is appropriate.

Definition VII.5.33 *If f is a recursive function and H is a partial recursive functional, then f is* **very *H*-honest** *if*

$$(\exists e)[f \simeq \varphi_e \ \wedge\ (\forall_\infty x)(H(\varphi_e, x)\!\downarrow \wedge\, (\max_{y\le x} \Phi_e(y)) \le H(\varphi_e, x))].$$

Thus, while the value $H(\varphi_e, x)$ almost always bounds $\Phi_e(x)$ for H-honest functions, it actually bounds $\max_{y\le x} \Phi_e(y)$ for very H-honest functions.

Theorem VII.5.34 Complexity-Theoretic Characterization of *EX* (Wiehagen and Liepe [1976], Wiehagen [1978]) *A class $\mathcal{C}$ of total recursive functions is in EX if and only if it is a class of very H-honest functions, for some partial recursive functional H.*

Proof. Given a partial recursive functional H, notice that if φ_e is total and very H-honest, then there is a constant k such that

$$(\forall x \ge k)[H(\varphi_e, x)\!\downarrow \Rightarrow\ (\max_{y\le x} \Phi_e(y)) \le H(\varphi_e, x)].$$

One can define a recursive function g that identifies by explanation the class of all total very H-honest functions, as follows:

- on the empty list, g takes the value 0
- on the list $\langle a_0, \ldots, a_n\rangle$, g takes the value e for the smallest pair $\langle e, k\rangle$ defined as follows, if there is one, and n otherwise:
  - $\langle e, k\rangle \le n$
  - for all x such that $k \le x \le n$, if $H(\langle a_0, \ldots, a_n\rangle, x)$ converges in at most n steps (where $\langle a_0, \ldots, a_n\rangle$ is considered as the partial function giving value a_x to $x \le n$ and undefined otherwise) then, for all $y \le x$,

$$\Phi_e(y) \le H(\langle a_0, \ldots, a_n\rangle, x) \qquad \text{and} \qquad \varphi_e(y) \simeq a_y.$$

g is total recursive because all checks are recursive.

It remains to show that g identifies by explanation the class of all total very H-honest functions. Suppose f is very H-honest. Then there is a pair $\langle e, k\rangle$ such that, for all $x \ge k$,

$$H(f, x)\!\downarrow, \qquad \max_{y\le x} \Phi_e(y) \le H(f, x) \qquad \text{and} \qquad \varphi_e(x) \simeq f(x),$$

and hence g must have a limit on f, since it must stop changing guess when it hits such a pair $\langle e, k\rangle$, if not before. Let then e be such a limit. Whenever $H(f, x)$ is defined, the construction ensures that $\varphi_e(y) \simeq f(y)$ for all $y \leq x$; since $H(f, x)$ is almost always defined, $\varphi_e \simeq f$.

In the opposite direction, let $\mathcal{C}$ be identifiable by explanation via g, and define

$$H(\alpha, x) \simeq \max\{\Phi_{g(\langle \alpha(0),\ldots,\alpha(x)\rangle)}(y) : y \leq x\}.$$

H is a partial recursive functional by definition.

Since g identifies $\mathcal{C}$, if $f \in \mathcal{C}$, then g has a limit e on f, and $\varphi_e \simeq f$. If x_0 is a point after which g does not change anymore on f, then for all $x \geq x_0$

$$H(f, x) \simeq \max\{\Phi_{g(\langle f(0),\ldots,f(x)\rangle)}(y) : y \leq x\} = \max_{y \leq x} \Phi_e(y).$$

Then e is a witness of the fact that f is very H-honest. $\square$

For other characterizations of EX, see Wiehagen and Liepe [1976], Wiehagen [1978], Klette and Wiehagen [1980], Freivalds, Kinber and Wiehagen [1982], Merkle and Stephan [1996].

The proof of VII.5.17 does not use any consistency hypothesis, and it thus shows that *the class of all total recursive functions is not in EX.*

Exercises VII.5.35 a) *The last observation is a corollary of VII.5.29.* (Hint: given an effectively separable r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$ with disagreement function d, define f total recursive not in it as follows. Having the values of f up to n, look for an e such that $\varphi_{h(e)}$ agrees with the given values up to n and is defined on all $x \leq d(e, x)$ (such an e must exist, if $\{\varphi_{h(e)}\}_{e\in\omega}$ contains all recursive functions). Then define $f(x) \simeq \varphi_{h(e)}(x)$ for all such x, so that f differs from $\varphi_{h(x)}$ on an argument $\leq d(e, x)$.)

b) *The last observation is a corollary of VII.5.31.* (Hint: given an effectively discrete r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$ with discreteness function d, choose any i. There are infinitely many recursive functions that agree with $\varphi_{h(i)}$ up to $d(i)$, but only finitely many can be in $\{\varphi_{h(e)}\}_{e\in\omega}$.)

c) *The last observation is a corollary of VII.5.34.* (Hint: see VII.5.24.)

Exercises VII.5.36 Finite identification. The next two notions formalize the idea of a method that waits until it can output the correct answer, and then does it.

A class $\mathcal{C}$ of total recursive functions is **finitely identifiable by next value** ($\mathcal{C} \in \boldsymbol{NV_{fin}}$) if it is identifiable by next value via a recursive function g for which there is a recursive procedure to determine, for any $f \in \mathcal{C}$, whether $g(\langle f(0), \ldots, f(n)\rangle)$ is correct from then on. (Klette [1976])

A class $\mathcal{C}$ of total recursive functions is **finitely identifiable by explanation** ($\mathcal{C} \in \boldsymbol{EX_{fin}}$) if it is identifiable by explanation via a recursive function g for which there is a recursive procedure to determine, for any $f \in \mathcal{C}$, whether $g(\langle f(0), \ldots, f(n)\rangle)$ has reached its limit. (Gold [1967])

An r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$ is **strongly effectively separable**, or **strongly effectively discrete**, if there is a uniform recursive procedure to determine, for every e, an upper bound to arguments on which $\varphi_{h(e)}$ disagrees with all other $\varphi_{h(i)}$, for $i \neq e$. (Wiehagen [1978])

a) $EX_{fin} - EX_{rel} \neq \emptyset$. (Bardzin) (Hint: consider the class $\mathcal{C} = \{f : f = \varphi_{f(0)}\}$, which is not in EX_{rel} by the proof of VII.5.27.)

b) $NV - EX_{fin} \neq \emptyset$. (Lindner) (Hint: consider the class $\mathcal{C} = \{f : f \simeq^* \mathcal{O}\}$, where $\mathcal{O}$ is the constant function with value 0. To show that $\mathcal{C} \notin EX_{fin}$, notice that if g identifies $\mathcal{C}$ by finite explanation then one establishes that $g(\langle f(0), \ldots, f(n)\rangle)$ is final solely on the basis of $f(0), \ldots, f(n)$, and thus $\mathcal{C}$ can contain only a single function with such an initial segment. This also leads to the characterization in part e).)

c) $EX_{fin} \cap NV \neq \emptyset$. (Hint: for a nontrivial example, i.e. an infinite class in the intersection, consider the class $\mathcal{C}$ of all polynomial functions with integer coefficients and of degree n, for a fixed n. Since each such function is determined by $n+1$ coefficients, $\mathcal{C} \in NV$. Since $n+1$ values are enough to determine the $n+1$ coefficients, $\mathcal{C} \in EX_{fin}$.)

d) *EX_{fin} is incomparable with all classes between NV and EX_{rel}*. (Hint: by parts a) and b).)

e) *A class $\mathcal{C}$ of total recursive functions is in EX_{fin} if and only if it is a subclass of a strongly effectively separable r.e. class of partial recursive functions*. This provides an analogue of both VII.5.29 and VII.5.31. (Wiehagen [1978]) (Hint: given a strongly effectively separable class $\{\varphi_{h(e)}\}_{e\in\omega}$, there is a recursive function d such that, for every different e and i, $\varphi_{h(e)}(x) \not\simeq \varphi_{h(i)}(x)$ for some $x \leq d(e)$. One can define a recursive function g that identifies by finite explanation any subclass $\mathcal{C}$ of the given class, as follows. On the empty list, g takes a value not in the range of h. On the list $\langle a_0, \ldots, a_n\rangle$, the value of g is equal to the value on the list $\langle a_0, \ldots, a_{n-1}\rangle$, unless there is an $e \leq n$ such that $d(e) \leq n$ and, for all $x \leq d(e)$, $\varphi_{h(e)}(x) \simeq a_x$ in at most n steps, in which case g takes its final value $h(e)$.

In the opposite direction, let $\mathcal{C}$ be identifiable by finite explanation via g. Consider the class of all functions

$$f_{\langle a_0,\ldots,a_n\rangle}(x) \simeq \begin{cases} a_x & \text{if } x \leq n \\ \varphi_{g(\langle a_0,\ldots,a_n\rangle)}(x) & \text{otherwise,} \end{cases}$$

for all sequence numbers of minimal length on which g has reached its limit. This class contains $\mathcal{C}$, and is strongly effectively separable via the function d that associates to each sequence number its length.)

f) *A class $\mathcal{C}$ of total recursive functions is in EX_{fin} if and only if there is a partial recursive functional F such that, for all $f \in \mathcal{C}$, $F(f)$ converges to an index of f*. This provides an analogue of VII.5.32. (Freivalds, Kinber and Wiehagen [1984]) (Hint: given g, let $F(\alpha)$ be the first value $g(\langle \alpha(0), \ldots, \alpha(n)\rangle)$ which is known to be final. Conversely, given F let $g(\langle a_0, \ldots, a_n\rangle)$ be $F(\langle a_0, \ldots, a_n\rangle)$, where $\langle a_0, \ldots, a_n\rangle$ is considered as a partial function, if it converges in at most n steps, and 0 otherwise.)

g) $EX_{fin} \cap NV = NV_{fin}$. (Hint: if $\mathcal{C} \in EX_{fin} \cap NV$, let g_1 be a guessing function for which one can decide whether the guess is final, and g_2 a next-value function. Define a next-value function for which one can decide when the values start being correct, by letting

$$g(\langle a_0,\ldots,a_n\rangle) = \begin{cases} g_2(\langle a_0,\ldots,a_n\rangle) & \text{if } g_1(\langle a_0,\ldots,a_n\rangle) \text{ is not yet final} \\ \varphi_{g_1(\langle a_0,\ldots,a_n\rangle)}(n+1) & \text{otherwise.} \end{cases}$$

Conversely, to show $NV_{fin} \subseteq EX_{fin}$, let g be a next-value function for which one can decide when the values start being correct, and define a guessing function g_1 for which one can decide whether the guess is final, by letting g_1 be an index for the completely undefined function until g starts giving correct values, and then letting g determine the correct program.)

For more on the present notion, see Wiehagen and Jung [1977], Wiehagen [1978], Freivalds and Wiehagen [1979], Freivalds, Kinber and Wiehagen [1982], [1984], Merkle and Stephan [1996].

Exercises VII.5.37 Identification by optimal explanation. A class of total recursive functions is **identifiable by optimal explanation** modulo a recursive factor ($\mathcal{C} \in \boldsymbol{EX_{opt}}$) if it is identifiable by explanation via a recursive function g such that, for some recursive function h and all $f \in \mathcal{C}$,

$$e \text{ minimal index of } f \ \Rightarrow\ (\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)) \leq h(e).$$

Thus, for any function in $\mathcal{C}$, g provides a program that is within the factor h from the minimal one. (Freivalds [1975])

a) $EX_{opt} - EX_{rel} \neq \emptyset$. (Chen [1982]) (Hint: consider $\mathcal{C} = \{f : f = \varphi_{f(0)}\}$, which is not in EX_{rel} by the proof of VII.5.27. To show $\mathcal{C} \in EX_{opt}$, let t be a recursive function such that $\varphi_{t(e)} \simeq \varphi_{\varphi_e(0)}$; define $h(e) = \max_{i\leq e} t(i)$, and g as the function that on the sequence number $\langle a_0,\ldots,a_n\rangle$ takes as value $t(i)$, for the least $i \leq n$ such that $\varphi_i(0) \simeq a_0$ in at most n steps (if there is no such i, then g takes the value 0).)

b) $NV - EX_{opt} \neq \emptyset$. (Kinber [1977]) (Hint: consider $\mathcal{C} = \{f : f \simeq^* \mathcal{O}\}$, where $\mathcal{O}$ is the constant function with value 0. To show that $\mathcal{C} \notin EX_{opt}$, suppose g is a recursive function identifying $\mathcal{C}$ by minimal programs, modulo the recursive factor h. The idea is to construct $f \in \mathcal{C}$ such that either $\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)$ does not exist (because g changes its guess infinitely often), or it exists but is not an index of f (because it differs from it on some arguments).

By the Fixed-Point Theorem, let e be an index of the function f to be constructed. We build f by finite initial segments σ_s, and let n_s be the greatest element on which σ_s is defined.

At stage $s+1$, if $g(\langle\sigma_s(0),\ldots,\sigma_s(n_s)\rangle)$ is $\leq h(e)$ and has already been cancelled, or it is $> h(e)$, then let $\sigma_{s+1}(n_s+1) = 0$, so that f obtains value 0 on a new argument. If $g(\langle\sigma_s(0),\ldots,\sigma_s(n_s)\rangle)$ is $\leq h(e)$ and has not been cancelled yet, see if either there is an extension of σ_s by 0's such that $g(\sigma) \neq g(\langle\sigma_s(0),\ldots,\sigma_s(n_s)\rangle)$, or there is an $x > n_s$ such that $\varphi_{g(\langle\sigma_s(0),\ldots,\sigma_s(n_s)\rangle)}(x)$ converges. In the first case, let $\sigma_{s+1} = \sigma$, thus making g change its value once. In the second case, cancel $g(\langle\sigma_s(0),\ldots,\sigma_s(n_s)\rangle)$ and extend σ_s to an initial segment σ_{s+1} such that

$$\sigma_{s+1}(x) \simeq 1 - \varphi_{g(\langle\sigma_s(0),\ldots,\sigma_s(n_s)\rangle)}(x),$$

thus making f different from the current guess.

If the construction stalls at stage $s+1$, it must be because $g(\langle \sigma_s(0), \ldots, \sigma_s(n_s)\rangle)$ is $\leq h(e)$ and has not been cancelled yet, there is no way of extending σ_s by 0's in such a way to make g change value, and $\varphi_{g(\langle \sigma_s(0), \ldots, \sigma_s(n_s)\rangle)}$ never converges for $x > n_s$. Let f be the extension of σ_s which is equal to 0 for all $x > n_s$. Then $f \in \mathcal{C}$,

$$g(\langle \sigma_s(0), \ldots, \sigma_s(n_s)\rangle) = \lim_{n\to\infty} g(\langle f(0), \ldots, f(n)\rangle),$$

and f is different from $\varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)}$ because the latter is a finite function.

If the construction never stalls, then it defines a total function f, which is in $\mathcal{C}$ because the only case in which the construction produces values different from 0 is when an index $\leq h(e)$ is cancelled, and this can happen only finitely often. If g reaches a limit on f, then it must be some index $\leq h(e)$ by the hypothesis on g. Since the construction does not stall, such a limit must eventually be cancelled. And since it cannot be cancelled by making g change value, it must be cancelled by making f different from the function guessed at that stage, and hence from the function guessed in the limit.)

c) $EX_{opt} \subset EX$. (Hint: by part b).)

For more on the present notion, see Freivalds [1975], [1990], and Chen [1982].

Identification by explanation with finite errors

There are two opposite ways of relaxing the requirements imposed in the definition of EX. We can be less restrictive on the function we identify in the limit and ask for an index not of the real f, but only of a finite approximation of it. Or we can be less restrictive on the convergence of our guesses and ask not for a fixed index in the limit, but only for indices of the same function. We investigate the first option here, and the second option in the next subsection.

Finitely many exceptions to the range of an explanation are readily accepted in science. For example, Newtonian mechanics was accepted even if it did not account correctly for the motion of Mercury's perihelion. The next notion is thus not without interest.

Definition VII.5.38 (Blum and Blum [1975]) *A class $\mathcal{C}$ of total recursive functions is* **identifiable by explanation with finitely many errors** *($\mathcal{C} \in \boldsymbol{EX^*}$) if there is a total recursive function g such that, for every $f \in \mathcal{C}$:*

- $\lim_{n\to\infty} g(\langle f(0), \ldots, f(n)\rangle)$ *exists*
- $\varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)} \simeq^* f$.

In other words, $g(\langle f(0), \ldots, f(n)\rangle)$ provides a guess to an index of f, and the guess stabilizes (from a certain point on) on an index of a finite variant of f.

Notice how, since we are now interested in guessing not the real function f but only a finite variant of it, we cannot request that the current guess agrees with the available information, and thus no analogue of EX_{cons} makes sense in this context.

Also, while in the definitions of EX_{cons} and EX the temporary guesses could be programs for partial functions but the final guesses had to be programs for a total function, here *all* guesses may actually be programs for partial functions (although the final guess must compute a function that can be undefined only on finitely many arguments, since it has to be a finite variant of a total functions).

Proposition VII.5.39 (Blum and Blum [1975]) $EX \subset EX^*$.

Proof. The inclusion is obvious by definition. To show that it is proper, we want to find a class $\mathcal{C}$ of total recursive functions which is in EX^* but not in EX. Let

$$\mathcal{C} = \{f : f \simeq^* \varphi_{f(0)}\},$$

i.e. the class of total recursive functions such that $f(0)$ is an index of a finite variant of f.

$\mathcal{C} \in EX^*$, by letting $g(\langle\ \rangle) = 0$ and $g(\langle a_0, \ldots, a_n\rangle) = a_0$. In other words, every function in $\mathcal{C}$ gives away a program for a finite variant of itself as its first value, thus making identification by explanation with finite errors trivial.

To show that $\mathcal{C} \notin EX$, suppose g is a recursive function such that

$$f \simeq^* \varphi_{f(0)} \Rightarrow f = \varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)}.$$

The idea is to construct $f \in \mathcal{C}$ such that either $\lim_{n\to\infty} g(\langle f(0), \ldots, f(n)\rangle)$ does not exist (because g changes its guess infinitely often), or it exists but it is not an index of f (being an index of a partial function).

There is no problem in forcing $f(0)$ to be an index of the function f to be defined, using the Fixed-Point Theorem as in the proof of VII.5.27. We thus concentrate on the definition of the remaining values of f. At each stage $n+1$, we have already defined all values of f up to n with one exception a_n, which is used to satisfy the idea above. There are three possible cases:

1. If $g(\langle f(0), \ldots, f(a_n-1)\rangle) \neq g(\langle f(0), \ldots, f(a_n-1), 0, f(a_n+1), \ldots, f(n)\rangle)$, then we define $f(a_n) = 0$, thus ensuring that g has changed its value once, and let $a_{n+1} = n+1$ (since now every value up to n has been defined).

2. If case 1 does not hold, and $\varphi_{g(\langle f(0),\ldots,f(a_n-1)\rangle)}(a_n)$ converges in at most n steps, then we define $f(a_n) = 1 - \varphi_{g(\langle f(0),\ldots,f(a_n-1)\rangle)}(a_n)$, thus ensuring that f is different from the function guessed on the initial segment of f of length a_n, and let $a_{n+1} = n+1$ as above.

3. If cases 1 and 2 do not hold, we let $f(n+1) = 0$, thus defining f on one more value, and $a_{n+1} = a_n$ (to have a new attempt at the next stage).

At the end, there are two possible cases:

- $\lim_{n\to\infty} a_n$ *does not exist*
  This means that we move a_n infinitely often, in particular the function f is total. Moreover, $f = \varphi_{f(0)}$ by construction, in particular $f \in \mathcal{C}$. Then, by hypothesis, $\lim_{n\to\infty} g(\langle f(0), \dots, f(n)\rangle)$ should exist and f should be $\varphi_{\lim_{n\to\infty} g(\langle f(0),\dots,f(n)\rangle)}$.

  But if $\lim_{n\to\infty} g(\langle f(0), \dots, f(n)\rangle)$ exists, then case 1 holds only finitely many times. Since a_n moves infinitely often, after a certain stage it must do so because of case 2. But then f is different from $\varphi_{\lim_{n\to\infty} g(\langle f(0),\dots,f(n)\rangle)}$ (because it disagrees with infinitely many guesses, and hence it must disagree with the final guess).

- $\lim_{n\to\infty} a_n$ *exists*
  If a is this limit, then f is defined everywhere except at a, since otherwise a would have moved as soon as $f(a)$ had been defined. Let f_1 be the extension of f obtained by letting $f_1(a) = 0$, and $f_1(x) = f(x)$ if $x \neq a$.

  Then $g(\langle f_1(0), \dots, f_1(a-1)\rangle) = \lim_{n\to\infty} g(\langle f_1(0), \dots, f_1(n)\rangle)$, otherwise case 1 would have taken place, and a would have moved.

  Moreover, $\varphi_{\lim_{n\to\infty} g(\langle f_1(0),\dots,f_1(n)\rangle)}(a)$ must be undefined, otherwise case 2 would have taken place, and a would have moved.

  $f_1(0) = f(0)$ is by definition an index of f, and hence of a finite variant of f_1. So $f_1 \simeq^* \varphi_{f_1(0)}$, but $f_1 \not\simeq \varphi_{\lim_{n\to\infty} g(\langle f_1(0),\dots,f_1(n)\rangle)}$, because the former is defined on a but the latter is not. $\square$

The proof of the previous result shows that actually

$$\mathcal{C} = \{f : f \simeq^* \varphi_{f(0)}, \text{ with at most one disagreement point}\}$$

is in $EX^* - EX$. In particular, even allowing for a *single* exception in the explanation of a class of phenomena already gives more power (in the sense of being able to explain more classes of phenomena) than requiring an explanation to be always correct.

Moreover, the proof also shows how the exception might occur in a place in which the explanation does not give any answer, being undefined. Since the Halting Problem is unsolvable (II.2.7), such an explanation (being correct when it does provide an answer, and incorrect in a divergent point) cannot be refuted (in a recursive way). In other words, *there are incomplete and irrefutable explanations*, and they do not satisfy a form of Popper's Refutability Principle (Popper [1934]), according to which incorrect scientific explanations should be refutable.

Exercise VII.5.40 *Both cases in the proof above must occur.* (Case and Smith [1983]) (Hint: let

$$\mathcal{C}_1 = \{f : f = \varphi_{f(0)}\}$$

and

$$\mathcal{C}_2 = \{f : f \simeq^* \varphi_{f(0)}, \text{ with exactly one disagreement point}\}.$$

Then

$$\mathcal{C}_1 \cup \mathcal{C}_2 = \{f : f \simeq^* \varphi_{f(0)}, \text{ with at most one disagreement point}\}.$$

Since $\mathcal{C}_1 \in EX$ by VII.5.27, and $\mathcal{C}_1 \cup \mathcal{C}_2 \notin EX$ by the proof of VII.5.39, it is enough to show that $\mathcal{C}_2 \in EX$. First, guess 0 as disagreement point, and as program one that assigns to 0 the value a_0, and to $x \neq 0$ the value $\varphi_{a_0}(x)$. If 0 really is the disagreement point, then this program is correct everywhere. If it is not, one will eventually discover it because then $\varphi_{a_0}(0) \simeq a_0$. If and when this happens, guess 1 as disagreement point, and as program one that assigns to 1 the value a_1, and to $x \neq 1$ the value $\varphi_{a_1}(x)$, and so on. Eventually one guesses as disagreement point the real one, and from then on the guess never changes. Thus $\mathcal{C}_2 \in EX$, because a disagreement point exists and is unique for any function in $\mathcal{C}$.)

Exercises VII.5.41 Identification by explanation with bounded errors (Case and Smith [1983]) A class of total recursive functions is in $\boldsymbol{EX^n}$ if it is **identifiable by explanation with at most $\boldsymbol{n}$ errors**. In particular, $EX^0 = EX$.

a) *For each n, $EX^n \subset EX^{n+1}$.* (Hint: consider the class of functions f such that $f \simeq^* \varphi_{f(0)}$, with at most $n+1$ disagreement points.)

b) $(\bigcup_{n\in\omega} EX^n) \subset EX^*$. (Hint: consider the class of functions f such that $f \simeq^* \varphi_{f(0)}$.)

We do not know of any number-theoretic characterization of EX^*. For a complexity-theoretic characterization, we already have at hand the appropriate notion.

Theorem VII.5.42 Complexity-Theoretic Characterization of $\boldsymbol{EX^*}$ (Wiehagen [1978], Kinber) *A class $\mathcal{C}$ of total recursive functions is in EX^* if and only if it is a class of H-honest functions, for some partial recursive functional H and w.r.t. space complexity measure.*

Proof. Given a partial recursive functional H, notice that if φ_e is H-honest, then there is a constant k such that

$$(\forall x \geq k)[H(\varphi_e, x)\downarrow \Rightarrow \Phi_e(x) \leq H(\varphi_e, x)].$$

One can define a recursive function g that identifies by explanation with finite errors the class of all total H-honest functions, as follows:

- on the empty list, g takes the value 0
- on the list $\langle a_0, \ldots, a_n\rangle$, g takes the value e for the smallest pair $\langle e, k\rangle$ defined as follows, if there is one, and n otherwise:
  - $\langle e, k\rangle \le n$
  - for all x such that $k \le x \le n$, if $H(\langle a_0, \ldots, a_n\rangle, x)$ converges in at most n steps (where $\langle a_0, \ldots, a_n\rangle$ is considered as the partial function giving value a_x to $x \le n$ and undefined otherwise), then

$$\Phi_e(x) \le H(\langle a_0, \ldots, a_n\rangle, x) \qquad \text{and} \qquad \varphi_e(x) \simeq a_x.$$

g is total recursive because all checks are recursive.

It remains to show that g identifies by explanation with finite errors the class of all total H-honest functions. Suppose f is an H-honest function. Then there is a pair $\langle e, k\rangle$ such that, for all $x \ge k$,

$$H(f, x)\!\downarrow, \quad \Phi_e(x) \le H(f, x) \qquad \text{and} \qquad \varphi_e(x) \simeq f(x),$$

and hence g must have a limit on f, since it must stop changing guesses when it hits such a pair $\langle e, k\rangle$, if not before. Let then e be such a limit. Whenever $H(f, x)$ is defined, the construction ensures that $\varphi_e(x) \simeq f(x)$; since $H(f, x)$ is almost always defined, φ_e is then a finite variant of f.

In the opposite direction, let $\mathcal{C}$ be identifiable by explanation with finite errors via g, and define

$$H(\alpha, x) \simeq \Phi_{g(\langle \alpha(0), \ldots, \alpha(x)\rangle)}(x).$$

H is a partial recursive functional by definition.

Since g identifies $\mathcal{C}$ with finite errors, if $f \in \mathcal{C}$, then g has a limit e on f, and $\varphi_e \simeq^* f$. If x_0 is a point after which g does not change anymore on f then, for all $x \ge x_0$,

$$H(f, x) \simeq \Phi_{g(\langle f(0), \ldots, f(x)\rangle)}(x) \simeq \Phi_e(x).$$

If moreover $x_1 \ge x_0$ is a point after which φ_e and f agree, then $\varphi_e(x)$ converges for all $x \ge x_1$, and hence

$$(\forall_\infty x)[H(f, x)\!\downarrow \wedge\, \Phi_e(x) \le H(f, x)].$$

We cannot claim that f is H-honest yet, since e is not an index of f (as required by Definition VII.5.22), but only of a finite variant of it. However, if we consider a complexity measure such as space, whose complexity classes are closed under finite variants (see VII.3.2.d), the result then follows. $\Box$

The proof of VII.5.17 uses functions that differ infinitely often, and thus shows that *the class of all total recursive functions is not in EX^*.*

Exercise VII.5.43 *The last observation is a corollary of VII.5.42.* (Hint: see VII.5.24.)

Behaviorally correct (consistent) identification

In the definitions of EX_{cons} and EX we required the final explanation of each phenomenon in a given class to be *intensionally* or *syntactically* unique. The next definition only asks for *extensional* or *semantical* uniqueness, and allows for the possibility of not having a final intensional explanation (i.e. it allows infinitely many changes in the program, although still only finitely many in the function defined by it).

Definition VII.5.44 (Feldman [1972], Bardzin [1974]) *A class $\mathcal{C}$ of total recursive functions is* **behaviorally correctly and consistently identifiable** *($\mathcal{C} \in \boldsymbol{BC_{cons}}$) if there is a total recursive function g such that, for every sequence number $\langle a_0, \dots, a_n\rangle$:*

- $\varphi_{g(\langle a_0,\dots,a_n\rangle)}(x) \simeq a_x$ *for all $x \leq n$,*

and for every $f \in \mathcal{C}$ and almost every n,

- $\varphi_{g(\langle f(0),\dots,f(n)\rangle)} = f.$

In other words, $g(\langle f(0), \dots, f(n)\rangle)$ provides a guess to an index of f consistent with the available information, and the guess stabilizes (from a certain point on) on indices of f.

$\mathcal{C}$ is **behaviorally correctly identifiable** *($\mathcal{C} \in \boldsymbol{BC}$) if the first condition on g is dropped.*

First, we see that the new notions of inference just introduced coincide, so that no analogue of VII.5.27 holds.

Proposition VII.5.45 $BC_{cons} = BC$.

Proof. $BC_{cons} \subseteq BC$ by definition. Conversely, if a procedure g outputs the guesses for functions in a class $\mathcal{C} \in BC$, then these guesses on $f \in \mathcal{C}$ are correct, and hence consistent in the limit. One can modify such a procedure, and output at stage n an index of the function that agrees with f on the arguments $\leq n$, and with $\varphi_{g(\langle f(0),\dots,f(n)\rangle)}$ otherwise. In other words, one outputs $g_1(\langle f(0), \dots, f(n)\rangle)$ for any g_1 such that

$$\varphi_{g_1(\langle a_0,\dots,a_n\rangle)}(x) \simeq \begin{cases} a_x & \text{if } x \leq n \\ \varphi_{g(\langle a_0,\dots,a_n\rangle)}(x) & \text{otherwise.} \end{cases}$$

If $f \in \mathcal{C}$, then $g(\langle f(0), \dots, f(n)\rangle)$ stabilizes on indices of f. Thus g_1 still identifies $\mathcal{C}$ in a behaviorally correct way, and is consistent by definition. Thus $\mathcal{C}$ is in BC_{cons}. $\square$

Exercises VII.5.46 a) *If in Definition VII.5.44 of BC the guesses are allowed to range (from a certain point on) only over a finite set of indices, then one obtains an alternative characterization of EX.* (Bardzin and Podnieks [1973], Case and Smith [1983]) (Hint: given $\langle a_0, \ldots, a_n\rangle$ consider all different values $g(\langle a_0, \ldots, a_m\rangle)$ for $m \leq n$, i.e. all different programs guessed so far, and make n steps in the computations of their values, for all arguments $x \leq n$. Discharge all programs that on some $x \leq n$ converge to a value different from a_x. Let $g_1(\langle a_0, \ldots, a_n\rangle)$ be a program for the function that, on any z, dovetails all programs guessed so far and not discharged, and outputs the first convergent value, if any. From a certain point on only finitely many programs will be considered, and they will all be correct on convergent computations, so that $g_1(\langle a_0, \ldots, a_n\rangle)$ reaches a limit.)

b) *If in Definition VII.5.1 of NV the next-value function is allowed to be partial, then one obtains an alternative characterization of BC.* (Podnieks [1974], [1975]) (Hint: if $\mathcal{C}$ is in BC via g, then let

$$\varphi(\langle a_0, \ldots, a_n\rangle) \simeq \varphi_{g(\langle a_0, \ldots, a_n\rangle)}(n+1).$$

Conversely, suppose φ is a partial recursive function such that, for every $f \in \mathcal{C}$ and almost every n,

$$f(n+1) \simeq \varphi(\langle f(0), \ldots, f(n)\rangle).$$

If g is such that

$$\varphi_{g(\langle a_0, \ldots, a_n\rangle)}(x) \simeq \begin{cases} a_x & \text{if } x \leq n \\ \varphi(\langle \varphi_{g(\langle a_0, \ldots, a_n\rangle)}(0), \ldots, \varphi_{g(\langle a_0, \ldots, a_n\rangle)}(x-1)\rangle) & \text{otherwise,} \end{cases}$$

then $\mathcal{C}$ is in BC via g.)

We now show that the new notion of inference is related to (and weaker than) those studied so far.

Proposition VII.5.47 (Steel) *$EX^* \subseteq BC$.*

Proof. If a procedure g outputs the guesses for functions in a class $\mathcal{C} \in EX^*$, then these guesses on $f \in \mathcal{C}$ are correct in the limit, but only for finite variants of f. One can modify such a procedure as in VII.5.45, by considering g_1 such that

$$\varphi_{g_1(\langle a_0, \ldots, a_n\rangle)}(x) \simeq \begin{cases} a_x & \text{if } x \leq n \\ \varphi_{g(\langle a_0, \ldots, a_n\rangle)}(x) & \text{otherwise.} \end{cases}$$

If $f \in \mathcal{C}$, then $g(\langle f(0), \ldots, f(n)\rangle)$ stabilizes on an index of a finite variant of f. Thus after a finite number of stages all modifications will compute the same function, and will be correct on all values. $\Box$

Notice that in the previous proof we cannot conclude $\mathcal{C} \in EX$, because if $f \in \mathcal{C}$, then $g_1(\langle f(0), \ldots, f(n)\rangle)$ codes a different program for every n, although a program obtained by patching up an eventually fixed program (for a finite variant of f) on an increasing number of arguments.

Proposition VII.5.48 (Bardzin [1974], Case and Smith [1983], Harrington) *$BC - EX^* \neq \emptyset$.*

Proof. We want to find a class of $\mathcal{C}$ of total recursive functions which is in BC but not in EX^*. Let

$$\mathcal{C} = \{f : (\forall_\infty x)(f = \varphi_{f(x)})\},$$

i.e. the class of total recursive functions such that $f(x)$ is an index of f, for almost every x.

$\mathcal{C} \in BC$, by letting $g(\langle\ \rangle) = 0$ and $g(\langle a_0, \dots, a_n\rangle) = a_n$. In other words, every function in $\mathcal{C}$ almost always gives away programs for itself as values, thus making behaviorally correct identification trivial.

To show that $\mathcal{C} \notin EX^*$, suppose g is a recursive function such that

$$(\forall_\infty x)(f = \varphi_{f(x)}) \ \Rightarrow\ f \simeq^* \varphi_{\lim_{n\to\infty} g(\langle f(0),\dots,f(n)\rangle)}.$$

The idea is to construct $f \in \mathcal{C}$ such that either $\lim_{n\to\infty} g(\langle f(0), \dots, f(n)\rangle)$ does not exist (because g changes its guess infinitely often), or it exists but it is not an index of f (being an index of a function which is not a finite variant of f).

We construct f by initial segments σ_s, starting from $\sigma_0 = \emptyset$, and let n_s be the greatest argument on which σ_s is defined. A natural strategy to satisfy the condition

$$f \not\simeq^* \varphi_{\lim_{n\to\infty} g(\langle f(0),\dots,f(n)\rangle)},$$

if the limit exists, is to make

$$f(x) \not\simeq \varphi_{g(\langle \sigma_s(0),\dots,\sigma_s(n_s)\rangle)}(x),$$

for a new x at every stage.

At stage $s+1$ we thus wait until $\varphi_{g(\langle \sigma_s(0),\dots,\sigma_s(n_s)\rangle)}(x)$ converges for some $x > n_s$, and then diagonalize. But since at the same time we want $f \in \mathcal{C}$, we will have to give f its own indices as values. We then choose two distinct indices e_0 and e_1 of f (by the Fixed-Point Theorem), and when $\varphi_{g(\langle \sigma_s(0),\dots,\sigma_s(n_s)\rangle)}(x)$ converges for some $x > n_s$ we extend σ_s by letting $\sigma_{s+1}(x) = e_i$, for an i (equal to 0 or 1) such that

$$e_i \neq \varphi_{g(\langle \sigma_s(0),\dots,\sigma_s(n_s)\rangle)}(x).$$

To have an initial segment, we also give all the arguments less than x and not in the domain of σ_s one of the values e_0 or e_1 (say, e_0).

Obviously, there is no guarantee that $\varphi_{g(\langle \sigma_s(0),\dots,\sigma_s(n_s)\rangle)}(x)$ will ever converge, for any $x > n_s$; and if it does not, then the construction would stall. We thus need an alternative strategy, and at stage $s+1$ we also build an additional function f_s, that will work if the construction of f stalls at that step.

Again, to ensure $f_s \in \mathcal{C}$ we will have to give f_s its own indices as values; we thus choose an index i_s of f_s (by the Fixed-Point Theorem), and at every step in the dovetailed computations of $\varphi_{g(\langle \sigma_s(0),\ldots,\sigma_s(n_s)\rangle)}(x)$ for $x > n_s$ we define f_s as i_s on a new argument, with the intent that if $\varphi_{g(\langle \sigma_s(0),\ldots,\sigma_s(n_s)\rangle)}(x)$ does not converge for any $x > n_s$, then f_s will be equal to i_s from a certain point on, and then automatically in $\mathcal{C}$.

This would not be of great help, unless we also ensured that

$$f_s \not\simeq^* \varphi_{\lim_{n\to\infty} g(\langle f_s(0),\ldots,f_s(n)\rangle)}.$$

We are working under the hypothesis that $\varphi_{g(\langle \sigma_s(0),\ldots,\sigma_s(n_s)\rangle)}(x)$ does not converge for any $x > n_s$, and hence that $\varphi_{g(\langle \sigma_s(0),\ldots,\sigma_s(n_s)\rangle)}$ is a finite function. It is thus natural to use this, by letting f_s extend σ_s; thus we will define f_s as equal to i_s only for $x > n_s$, and equal to $\sigma_s(x)$ if $x \leq n_s$. If

$$g(\langle \sigma_s(0),\ldots,\sigma_s(n_s)\rangle) = \lim_{n\to\infty} g(\langle f_s(0),\ldots,f_s(n)\rangle),$$

then

$$f_s \not\simeq^* \varphi_{\lim_{n\to\infty} g(\langle f_s(0),\ldots,f_s(n)\rangle)}$$

because f_s is total, while the right hand side is a finite function.

Obviously, there is no guarantee that

$$g(\langle \sigma_s(0),\ldots,\sigma_s(n_s)\rangle) = \lim_{n\to\infty} g(\langle f_s(0),\ldots,f_s(n)\rangle).$$

But if this does not hold, then g changes value on f_s sooner or later, and we can take advantage of this to go back to the definition of f. Indeed, by the first strategy we tried to make $\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)$ not an index of f, while with the present back up strategy we are trying to ensure that the limit does not exist.

In other words, we continue to define f_s as i_s for more and more arguments, until we discover that for an initial segment τ of f_s,

$$g(\tau) \neq g(\langle \sigma_s(0),\ldots,\sigma_s(n_s)\rangle).$$

If this happens, then we let $\sigma_{s+1} = \tau$. This is consistent with what was previously done, i.e. $\sigma_{s+1} \supseteq \sigma_s$, because f_s extends σ_s.

Notice that defining $\sigma_{s+1} = \tau$ gives f value i_s for a number of arguments; to avoid ruining the strategy for $f \in \mathcal{C}$, we better make i_s an index of f. Thus, we stop defining new values of f_s as i_s (since, in any case, f_s has lost its role as possible witness of the fact that g does not identify $\mathcal{C}$), and from now on the definition of f_s will just copy the definition of f.

We now have to argue that the proposed construction works. There are two possible cases:

- *all stages terminate*

  Each value of f is either one of e_0 and e_1, or some i_s; hence an index of f, either by the initial choice of e_0 and e_1, or by construction of f_s. In particular, $(\forall x)(f = \varphi_{f(x)})$, and $f \in \mathcal{C}$.

  If g has no limit on f, there is nothing to prove. If g does have a limit on f, then we look at the construction of f. Every time the second part of the construction is applied at some stage, g changes value on f; since g has a limit on f by hypothesis, the second part of the construction can thus by applied only finitely many times. Then the first part of the construction is applied almost always; but every time it is applied, a disagreement between f and the current guess is enforced, and once the guess has stabilized, a disagreement with the final guess is enforced. In particular, $f \not\simeq^* \varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)}$.

- *some stage $s+1$ does not terminate*

  By construction, f_s is then total. Moreover, almost all of its values are i_s, hence an index of f_s by the choice of i_s. In particular, $(\forall_\infty x)(f_s = \varphi_{f_s(x)})$, and $f_s \in \mathcal{C}$.

  By construction

  $$g(\langle \sigma_s(0),\ldots,\sigma_s(n_s)\rangle) = \lim_{n\to\infty} g(\langle f_s(0),\ldots,f_s(n)\rangle),$$

  otherwise stage $s+1$ would terminate by the second part of the construction. Moreover, $\varphi_{g(\langle \sigma_s(0),\ldots,\sigma_s(n_s)\rangle)}$ is a finite function, because if it were defined on some $x > n_s$, then stage $s+1$ would terminate by the first part of the construction. In particular, $f_s \not\simeq^* \varphi_{\lim_{n\to\infty} g(\langle f_s(0),\ldots,f_s(n)\rangle)}$, because f_s is total.

We have made infinitely many appeals to the Fixed-Point Theorem in the constructions of f and of the f_s (which is not surprising, since f had to be self-referential infinitely often, to be in $\mathcal{C}$). There would be no problem if these constructions were independent, but they are instead related one to the others (f may use the index i_s, and f_s may mimic f). We thus still have to make sure that it is really possible to construct an infinite sequence of functions, simultaneously using indices for all of them.

Since a sequence of indices can be thought of as the range of a function f, and the previous construction actually produces a recursive functional F, the next result provides a formal justification.

- **Functional Recursion Theorem (Case [1974a])** *Given a partial recursive functional F, there is a total recursive function f such that*

$$(\forall e)(\forall x)[\varphi_{f(e)}(x) \simeq F(f,e,x)].$$

Proof. It is enough to find a recursive function g such that

$$\varphi_{\varphi_{g(i)}(e)}(x) \simeq F(\varphi_i, e, x).$$

Then the usual Fixed-Point Theorem gives an i such that $\varphi_i \simeq \varphi_{g(i)}$, and

$$\varphi_{\varphi_i(e)}(x) \simeq \varphi_{\varphi_{g(i)}(e)}(x) \simeq F(\varphi_i, e, x).$$

By taking $f \simeq \varphi_i$, we obtain the result.

To obtain g, consider the partial recursive function

$$\psi(i, e, x) \simeq F(\varphi_i, e, x).$$

By the S_n^m-Theorem, there is a recursive function h such that

$$\varphi_{h(i,e)}(x) \simeq \psi(i, e, x).$$

By the S_n^m-Theorem again, there is a recursive function g such that

$$\varphi_{g(i)}(e) \simeq h(i, e).$$

Then

$$\varphi_{\varphi_{g(i)}(e)}(x) \simeq \varphi_{h(i,e)}(x) \simeq \psi(i, e, x) \simeq F(\varphi_i, e, x).$$

Notice that h is total because defined by the S_n^m-Theorem (see II.1.7), and hence so are $\varphi_{g(i)}$ and φ_i, i.e. f. □

We do not know of any complexity-theoretic characterization of BC. The next result provides a number-theoretic one in the style of the characterizations VII.5.29 of EX and VII.5.36.e of EX_{fin}.

Definition VII.5.49 *Given an r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$ of partial recursive functions, a subclass $\mathcal{C}$ of total recursive functions is* **weakly effectively discrete** *in it if there is a uniform recursive procedure to determine, for every $f \in \mathcal{C}$ and almost every i, an upper bound to arguments on which f disagrees with $\varphi_{h(i)}$, if they disagree at all.*

The notion just introduced is a double weakening of effective discreteness. First, it allows for infinitely many repetitions in the given class, while only finitely many were allowed in VII.5.30. Second, it is not a global condition on the given class, but rather a local one on the given subclass.

Theorem VII.5.50 Number-Theoretic Characterization of BC (Wiehagen) *A class $\mathcal{C}$ of total recursive functions is in BC if and only if it is a weakly effectively discrete subclass of an r.e. class of partial recursive functions.*

Proof. Given a class $\mathcal{C}$ weakly effectively discrete in an r.e. class $\{\varphi_{h(e)}\}_{e\in\omega}$, there is a recursive function d such that, for every $\varphi_{h(e)} \in \mathcal{C}$, there are at most finitely many i such that if $\varphi_{h(e)}(x) \simeq \varphi_{h(i)}(x)$ for all $x \leq d(e)$, then $\varphi_{h(e)}$ and $\varphi_{h(i)}$ are different. One can define a recursive function g that behaviorally correctly identifies $\mathcal{C}$ as in the proof of VII.5.31.

Indeed, given any $f \in \mathcal{C}$, the definition of g and the fact that $\mathcal{C}$ is weakly effectively discrete in $\{\varphi_{h(e)}\}_{e\in\omega}$ imply that for any sufficiently large n the sets $I_{\langle f(0),\ldots,f(n)\rangle}$ contain at least one index of f, and only indices of functions compatible with it. From that point on, $g(\langle f(0),\ldots,f(n)\rangle)$ will always be an index of f. The only difference with the proof of VII.5.31 is that we can no longer claim that the sets $I_{\langle f(0),\ldots,f(n)\rangle}$ reach a finite limit, and hence that so does $g(\langle f(0),\ldots,f(n)\rangle)$.

In the opposite direction, let $\mathcal{C}$ be behaviorally correctly identifiable via g. Then every function in $\mathcal{C}$ is of the following form, for some sequence number a (coding a list $\langle a_0,\ldots,a_n\rangle$ for some n):

$$f_a(x) \simeq \begin{cases} a_x & \text{if } x \leq n \\ \varphi_{g(\langle a_0,\ldots,a_n\rangle)}(x) & \text{otherwise} \end{cases}$$

(more precisely, $f \in \mathcal{C}$ is equal to f_a for any sequence number a coding the first n values of f, for any n greater than the point after which g only outputs indices of f). Any such f_a is partial recursive uniformly in a, and by the S^m_n-Theorem there is then a recursive function h such that $\varphi_{h(a)} = f_a$. Thus the class $\{f_a\}_{a\in\omega}$ is an r.e. class of partial recursive functions containing $\mathcal{C}$.

Let $d(a)$ be the length of a. Suppose that $f \in \mathcal{C}$ agrees with f_a up to $d(a)$, i.e. a is an initial segment of f. If $d(a)$ is greater than the point after which g only outputs indices of f, which must happen for almost all a, then $f_a = f$ by definition. □

The proof of VII.5.17 does not automatically show that the class of all total recursive functions is not in BC (as it did for EX and EX^*), because it forces changes only in the guessed programs, as opposed to the guessed functions. But a simple modification of it will do.

Proposition VII.5.51 (Bardzin [1974], Case and Smith [1983]) *The class of all total recursive functions is not in BC.*

Proof. Let g be a recursive function such that

$$f \text{ total recursive} \Rightarrow \lim_{n\to\infty} \varphi_{g(\langle f(0),\ldots,f(n)\rangle)} = f.$$

We define a total recursive function f such that $\lim_{n\to\infty} \varphi_{g(\langle f(0),\ldots,f(n)\rangle)}$ does not exist, contradicting the hypothesis on g. The idea is that f will repeat the same value a number of times sufficient to make g converge, and then will diagonalize against the program thus obtained. Since we do this infinitely often, then the function defined by g changes infinitely often on f, and thus it has no limit.

Start with $f(0) = 0$. The function f_0 identically equal to 0 is recursive and, by hypothesis on g, $\lim_{n\to\infty} \varphi_{g(\langle f_0(0),\ldots,f_0(n)\rangle)}$ exists. Thus there must exist n and $m > n$ such that $\varphi_{g(\langle f_0(0),\ldots,f_0(n)\rangle)}(m)\downarrow$, and we can effectively find them (by dovetailing computations). Let

$$f(x) = \begin{cases} f_0(x) & \text{if } x \le n \\ 0 & \text{if } n \le x < m \\ \varphi_{g(\langle f_0(0),\ldots,f_0(n)\rangle)}(x) + 1 & \text{if } x = m, \end{cases}$$

and iterate the procedure (by considering, at each step s, the function f_s extending the values already obtained by a sequence of 0's). $\square$

Exercise VII.5.52 *VII.5.51 is a corollary of VII.5.50.* (Hint: see VII.5.35.b.)

For other characterizations of BC, see Merkle and Stephan [1996].

Behaviorally correct identification with finite errors

We next relax BC in the same as we did for EX in VII.5.38.

Definition VII.5.53 (Osherson and Weinstein [1982], Case and Smith [1983]) *A class $\mathcal{C}$ of total recursive functions is* **behaviorally correctly identifiable with finitely many errors** *($\mathcal{C} \in$ **BC^***) if there is a recursive function g such that, for every $f \in \mathcal{C}$ and almost every n,*

$$\varphi_{g(\langle f(0),\ldots,f(n)\rangle)} \simeq^* f.$$

In other words, $g(\langle f(0),\ldots,f(n)\rangle)$ provides a guess to an index of f, and the guess stabilizes (from a certain point on) on indices of finite variants of f.

Notice that BC^* is a very weak notion. Not only the guesses can change infinitely often, as they did for BC, but even the functions that they compute can do so. The only requirement is that each such function eventually be a finite variant of f.

Exercises VII.5.54 a) *If in Definition VII.5.53 of BC^* the guesses are allowed to range (from a certain point on) only over a finite set of indices, then one obtains an alternative characterization of EX^*.* (Case and Smith [1983]) (Hint: as in VII.5.46.a, with the difference that on $\langle a_0,\ldots,a_n\rangle$ one discharges the programs such that the number of $x \le n$ on which they converge to a value different from a_x is greater than the number of times they have been output until then. From a certain point on only finitely many programs will be considered, and they will all be correct on convergent computations, with at most finitely many exceptions each.)

b) *If in Definition VII.5.53 of BC^* the guesses are allowed to range (from a certain point on) only over a finite set of functions, then one obtains an alternative characterization of BC.* (Hint: as in VII.5.45 and VII.5.47.)

The next result shows that we have finally reached the possible limits of inductive inference.

Theorem VII.5.55 Characterization of BC^* (Harrington) *The class of all (and hence any class of) total recursive functions is in BC^*.*

Proof. Let $\mathcal{C}$ be the class of all total recursive functions. Notice that we could easily identify $\mathcal{C}$ by consistent explanation, if we contented ourselves with a guessing function recursive in $\mathcal{K}$. In this case we could just let g be the function defined as in the proof of VII.5.9, as follows:

- on the empty list, g takes the value 0;
- on the list $\langle a_0, \ldots, a_n \rangle$, g takes the value e, for the first e such that $\varphi_e(x) \simeq a_x$ for all $x \leq n$ (i.e. g guesses the first partial recursive function that converges and agrees with all values coded by the given list).

Obviously, g is a function recursive in $\mathcal{K}$. Given an a such that $g = \varphi_a^{\mathcal{K}}$, by the S_n^m-Theorem there is a recursive function g_1 such that

$$\varphi_{g_1(\langle a_0,\ldots,a_n\rangle)}(x) \simeq \varphi_{\varphi_a^{\mathcal{K}_x}(\langle a_0,\ldots,a_n\rangle)}(x).$$

We now prove that $\mathcal{C} \in BC^*$ via g_1.

Let f be a total recursive function. We show that, for any sufficiently large n, $g_1(\langle f(0), \ldots, f(n)\rangle)$ is an index of a finite variant of f. First, let n_0 be large enough so that g has reached its final value e on $\langle f(0), \ldots, f(n_0)\rangle$. Second, given $n \geq n_0$, let x_n be large enough so that $\varphi_a^{\mathcal{K}_{x_n}}$ has reached its final value on $\langle f(0), \ldots, f(n)\rangle$. In other words, let n_0 be a bound on the information on f needed to compute the correct index recursively in $\mathcal{K}$, and let x_n be a bound on the information on $\mathcal{K}$ needed to compute a correct approximation recursively. Then, for every $x \geq x_n$,

$$\varphi_{g_1(\langle f(0),\ldots,f(n)\rangle)}(x) \simeq \varphi_e(x) \simeq f(x),$$

i.e. $g_1(\langle f(0), \ldots, f(n)\rangle)$ is an index of a finite variant of f. $\Box$

Notice that we only proved that

$$(\exists n_0)(\forall n \geq n_0)(\exists x_n)(\forall x \geq x_n)[\varphi_{g_1(\langle f(0),\ldots,f(n)\rangle)}(x) \simeq f(x)],$$

because $g_1(\langle f(0),\ldots,f(n)\rangle)$ possibly codes a different finite variant of f for every n, being correct (for $n \geq n_0$) only for $x \geq x_n$, where x_n depends on n. In particular, we cannot conclude that $\mathcal{C} \in BC$ by VII.5.54.b, i.e. that

$$(\exists n_0)(\exists x_0)(\forall n \geq n_0)(\forall x \geq x_0)[\varphi_{g_1(\langle f(0),\ldots,f(n)\rangle)}(x) \simeq f(x)].$$

Corollary VII.5.56 *$BC \subset BC^*$.*

Proof. The inclusion is obvious by definition. That it is proper follows from VII.5.51 and VII.5.55. $\square$

Exercises VII.5.57 Behaviorally correct identification with bounded errors. (Case and Smith [1983]) A class of total recursive functions is in $\boldsymbol{BC^n}$ if it is **behaviorally correctly identifiable with at most $\boldsymbol{n}$ errors**. In particular, $BC^0 = BC$.

a) *For each n, $BC^n \subset BC^{n+1}$.* (Hint: consider the class of functions f such that, for almost every x, $f \simeq \varphi_{f(x)}$, with at most $n+1$ disagreement points.)

b) $(\bigcup_{n\in\omega} BC^n) \subset BC^*$. (Hint: consider the class of functions f such that, for almost every x, $f \simeq^* \varphi_{f(x)}$.)

Exercises VII.5.58 Closure under union. The interest of closure under union is that makes it possible to use previous experience to learn more, thus providing some kind of cumulative learning.

a) *NV is closed under union.* (Hint: by VII.5.2, since the union of two r.e. classes of functions is still r.e.)

b) *EX_{cons} is closed under union.* (Viviani) (Hint: by VII.5.14, since the union of two quasi-measured classes of functions is still quasi-measured.)

c) *EX_{rel} is closed under union.* (Minicozzi [1976]) (Hint: let g_1 and g_2 identify by reliable explanation $\mathcal{C}_1$ and $\mathcal{C}_2$. We can suppose that, for $i = 0, 1$,

$$g_i(\langle a_0,\ldots,a_n\rangle) \neq g_i(\langle a_0,\ldots,a_{n+1}\rangle) \;\Rightarrow\; g_i(\langle a_0,\ldots,a_{n+1}\rangle) \geq n.$$

Define g as follows:

$$g(\langle a_0,\ldots,a_n\rangle) = \min\{g_1(\langle a_0,\ldots,a_n\rangle), g_2(\langle a_0,\ldots,a_n\rangle).\}$$

Then g identifies $\mathcal{C}_1 \cup \mathcal{C}_2$ by reliable explanation.)

d) *EX is not closed under union.* (Bardzin [1974], Blum and Blum [1975]) (Hint: see VII.5.40. More directly, let

$$\mathcal{C}_1 = \{f : \varphi_{f(0)}\} \qquad \text{and} \qquad \mathcal{C}_2 = \{f : f \simeq^* \mathcal{O}\},$$

where $\mathcal{O}$ is the constant function with value 0. Both $\mathcal{C}_1$ and $\mathcal{C}_2$ belong to EX. To show that $\mathcal{C}_1 \cup \mathcal{C}_2 \notin EX$, suppose there is g such that

$$f = \varphi_{f(0)} \text{ or } f \simeq^* \mathcal{O} \quad \Rightarrow \quad f = \varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle)}.$$

By the Fixed Point Theorem, we can define $f(0)$ in such a way that $f = \varphi_{f(0)}$. It is then enough to force f to differ from $\varphi_{\lim_{n\to\infty}(\langle f(0),\ldots,f(n)\rangle)}$, and this can be done as in VII.5.51.)

e) *EX^* is not closed under union.* (Bardzin [1974], Blum and Blum [1975]) (Hint: see part d).)

f) *BC is not closed under union.* (Bardzin [1974], Case and Smith [1983]) (Hint: see part d).)

g) *BC^* is closed under union.* (Hint: by VII.5.55.)

The example of part d) is obviously learnable by a *team* of two machines, and suggests the possibility of accordingly extending the notion of identification by explanation. For more on this, see Smith [1982], and Pitt and Smith [1988].

The world of inference classes ⋆

The sequence of inclusions and equalities proved in VII.5.9, VII.5.11, VII.5.20, VII.5.21, VII.5.27, VII.5.39, VII.5.45, VII.5.47, VII.5.48, and VII.5.56 provide the following picture:

$$NV \subset EX_{cons} \subset EX_{rel} \subset EX \subset EX^* \subset BC_{cons} = BC \subset BC^*.$$

The picture is refined by VII.5.41 and VII.5.57, which provide the following infinite sequences of classes:

$$EX \subset EX^1 \subset EX^2 \subset \cdots \subset \bigcup_{n\in\omega} EX^n \subset EX^*$$

and

$$BC \subset BC^1 \subset BC^2 \subset \cdots \subset \bigcup_{n\in\omega} BC^n \subset BC^*.$$

Degrees of inferability ⋆

The theory of Inductive Inference just developed can be relativized to any oracle A, by using in the various definitions functions g that are recursive in A, and not only recursive (however, the functions to be inferred are still supposed to be recursive).

The relativization of EX to A is denoted by EX^A. We can compare the relative power of oracles w.r.t. EX by letting

$$A \leq_{EX} B \Leftrightarrow EX^A \subseteq EX^B.$$

$\leq_{EX}$ is a reflexive and transitive relation, that induces an equivalence relation

$$A \equiv_{EX} B \ \Leftrightarrow \ A \leq_{EX} B \ \wedge \ B \leq_{EX} A.$$

The equivalence classes of sets w.r.t. $\equiv_{EX}$ are called ***EX*-degrees**, and $\boldsymbol{\mathcal{D}_{EX}}$ is the structure of EX-degrees, with the partial ordering induced on them by $\leq_{EX}$.

$\boldsymbol{\mathcal{D}_{EX}}$ obviously has a *least element* $\mathbf{0}_{EX}$, containing exactly the sets A such that $EX = EX^A$. It also has a *greatest element* $\mathbf{1}_{EX}$, containing exactly the sets A such that the class of all recursive functions is in EX^A. Such oracles exist by the proof of VII.5.55, where it is shown that recursively in $\emptyset'$ we can infer all total recursive functions. The existence of a greatest EX-degree shows that $\boldsymbol{\mathcal{D}_{EX}}$ *is not elementarily equivalent to any of the usual degree structures.*

Since we can infer all total recursive functions recursively in any A such that $\emptyset' \leq_T A$, $\mathbf{1}_{EX}$ *contains uncountably many sets*, again unlike any usual degree structure. A complete characterization of $\mathbf{1}_{EX}$, as well as of $\mathbf{0}_{EX}$, is the following:

- $A \in \mathbf{0}_{EX}$ *if and only if* $A \leq_T \emptyset'$ *and it is either recursive or 1-generic* (Slaman and Solovay [1991])
- $A \in \mathbf{1}_{EX}$ *if and only if* $\emptyset'' \leq_T A'$ (Adleman and Blum [1991]).

It follows in particular that the join operation is not well-defined on EX-degrees, since $\mathcal{K}$ can be split as the join of two 1-generic sets (V.2.26). It is not known whether $\boldsymbol{\mathcal{D}_{EX}}$ is an uppersemilattice.

Kummer and Stephan [1996] have proved that any EX-degree different from $\mathbf{1}_{EX}$ contains only sets having the same jump, and hence only countably many sets, so that $\boldsymbol{\mathcal{D}_{EX}}$ *has cardinality* $2^{\aleph_0}$. Moreover, *there are minimal* EX*-degrees.*

Since, obviously,

$$A \leq_T B \ \Rightarrow \ A \leq_{EX} B,$$

every EX-degree consists of T-degrees. Kummer and Stephan [1996] have proved that *every* EX*-degree contains infinitely many* T*-degrees.*

One can also study the substructure $\boldsymbol{\mathcal{R}_{EX}}$ of the r.e. EX-degrees. In this case $\mathbf{0}_{EX}$ consists exactly of the recursive sets (by XI.2.3.3, no 1-generic set has r.e. degree), and $\mathbf{1}_{EX}$ consists exactly of the high r.e. degrees. Kummer and Stephan [1996] have proved that, for A and B r.e. and not high,

$$A \leq_{EX} B \ \Leftrightarrow \ A \leq_T B.$$

Thus $\boldsymbol{\mathcal{R}_{EX}}$ *is the structure obtained from* $\boldsymbol{\mathcal{R}}$ *by collapsing all the high r.e. degrees into a single degree.*

More results about $\boldsymbol{\mathcal{D}_{EX}}$, as well as about degree structures similarly defined, such as $\boldsymbol{\mathcal{D}_{NV}}$ and $\boldsymbol{\mathcal{D}_{BC}}$, are in Gasarch and Pleszkoch [1989], Fortnow, Jain, Gasarch, Kinber, Kummer, Kurtz, Pleszkoch, Slaman, Stephan and Solovay [1994], Kummer and Stephan [1996].

æ

Chapter VIII

Hierarchies of Recursive Functions

The present chapter is devoted to a synthetic (bottom-up) study of the class of recursive functions that was introduced analytically in Chapter I. We attempt a construction of such a class from below, by increasingly more comprehensive subclasses (the historical development proceeded backwards, by increasingly more restricted subclasses). Although, as discussed on p. 60, there can be no natural exhaustive hierarchy for the recursive functions, we introduce and study a number of interesting partial hierarchies.

In Sections 1 to 6 we apply the **computational complexity theory** developed in Chapter VII to the study of complexity classes defined by various time and space bounds. More precisely, we deal with *small time and space bounds* in Section 1, deterministic and nondeterministic *polynomial time* in Sections 2–4, *polynomial space* in Section 5 and *exponential time and space* in Section 6.

In Sections 7 to 9 we turn instead to **number theory** and study classes of recursive functions defined by various algebraic closure operations and proof-theoretical versions of recursion. More precisely, we deal with *elementary* functions in Section 7, *primitive recursive* functions in Section 8, and ϵ_0-*recursive* functions in Section 9.

In assigning names to classes of recursive functions, we adopt the following **convention**:

- **BOUND** and **BOUNDSPACE** indicate corresponding classes defined by the same time and space bound, respectively.

- **CLASS** and **NCLASS** indicate corresponding classes defined by deter-

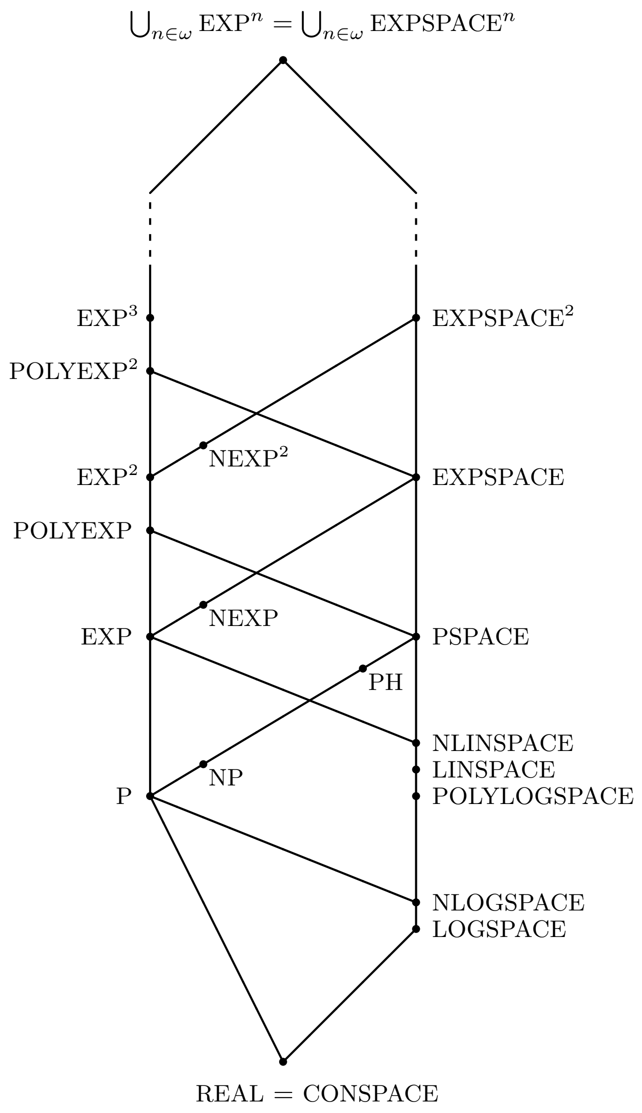


Figure VIII.1: Time and Space Complexity Classes

ministic and nondeterministic bounds, respectively

- **CLASS** and **CLASSF** indicate corresponding classes of sets and functions, respectively
- **co-CLASS** indicates the class of complements of sets in **CLASS**.

VIII.1 Small Time and Space Bounds ⋆

The problem of a synthesis of the class of recursive functions has already been discussed in Chapter VII from an abstract point of view. In particular, a number of general notions and tools useful in the construction of hierarchies of recursive functions have been introduced in Sections VII.1–3. The level of abstraction has been drastically reduced in Section VII.4, with a shift of attention to the specific measures of Turing machine time and space. We now start a detailed investigation of complexity classes w.r.t. such measures, by looking at small bounds.

Real time

From the Union Theorem VII.3.7, we know that there is a gap at the lower end of the time complexity classes, consisting of the *functions computable in constant time*. This class is not too interesting, since it depends heavily on details of the machine model. For example, if unary notation is used for the input then only functions eventually constant are computable in constant time, since almost all inputs are indistinguishable. On the other hand, if binary notation is used for the input and at the beginning the head is positioned, for example, on the rightmost cell of the input, then the machine can detect in constant time whether the input is even or odd (by checking whether the last digit is 0 or 1). But this is not possible if at the beginning the head is positioned on the leftmost cell of the input, since to move to the rightmost cell requires travelling through the input.

If a function is not computable in constant time, then its values must depend on x for infinitely many x. The next observation provides the next gap at the lower end of the time complexity classes.

Proposition VIII.1.1 *If a function f is not computable in constant time, then any Turing machine computing it must take at least time $|x|$ for infinitely many x.*

Proof. Given f not computable in constant time, let M be a Turing machine computing f in time $t(x)$. Since f is not computable in constant time, there is an infinite sequence of inputs x_n such that

$$t(x_0) < t(x_1) < \cdots$$

If we let x_n^* be equal to x_n if $t(x_n) \geq |x_n|$, and equal to the initial segment of x_n of length $t(x_n)$ otherwise, then M gives the same output in the same time on x_n and x_n^*, i.e. $f(x_n^*) = f(x_n)$ and $t(x_n^*) = t(x_n) \geq |x_n^*|$. Thus M takes at least time $|x|$ for infinitely many x. $\square$

Thus, the next time complexity class after the class of functions computable in constant time is

$$\mathrm{REAL(F)} = \mathrm{TIME(F)}\,[\,|x|\,],$$

consisting of the sets (functions) computable in **real time**. There are indeed (0,1-valued) functions computable in real but not in constant time, and hence the result cannot be improved.

The notion of real time was introduced by Yamada [1962] and it again depends heavily on details of the machine model (see Wagner and Wechsung [1986] for a survey of results). For example, Rabin [1963] and Anderaa [1974] have shown that, for each $n \geq 1$, there are functions computable in real time by a Turing machine with $n+1$ tapes, but not computable in real time by any Turing machine with n tapes (see VII.4.7.a for the case $n = 1$, and Paul, Seiferas and Simon [1981] for a simplified proof of the general case).

For the one-tape Turing machine model we are using, the class of sets computable in real time coincides with the class of sets computable in constant space studied in the next subsection (see VIII.1.7).

Exercise VIII.1.2 $\{e : M_e$ *works in real time*$\}$ *is* Π_1^0*-complete.* (Hájek [1979]) (Hint: it is Π_1^0 because

$$M_e \text{ works in real time } \Leftrightarrow\ (\forall x)(\varphi_e(x)\!\downarrow\, \Rightarrow\ T_e(x) \leq |x|).$$

To show Π_1^0-completeness, define a recursive function f such that

$$e \in \overline{\mathcal{K}}\ \Leftrightarrow\ M_{f(e)} \text{ works in real time}$$

as follows. On input x, let $M_{f(e)}$ simulate M_e on input e for $|x|$ steps. If M_e has not yet halted, $M_{f(e)}$ halts then. Otherwise, $M_{f(e)}$ halts after one more step.)

We now look beyond real time. The next result shows that nothing is gained, unless at least $|x| \cdot \log |x|$ moves are allowed.

Proposition VIII.1.3 (Trakhtenbrot [1964], Hartmanis [1968]) *If a function is not computable in time $|x|$, then any Turing machine computing it must take at least time proportional to $|x| \cdot \log |x|$ for infinitely many x.*

Proof. Given f not computable in time $|x|$, let M be a Turing machine (with m states) computing f in time $t(x)$, and let $cr(x)$ be the maximum number of crossings between two adjacent cells of the input x during the computation of M.

Since f is not computable in time $|x|$, $cr(x)$ cannot be bounded by a constant. Since $cr(x)$ is unbounded, given n there is an input x_n of smallest length such that $cr(x_n) \geq n$.

The computation of the lower bound in the statement of the proposition rests on the following fact, whose proof uses the notion of a crossing sequence introduced on p. 73:

- *every crossing sequence of M on x_n can occur at most twice on cells of the input x_n*
  Otherwise, if the same crossing sequence occurred at least three times, one could split the input x_n into four nonempty parts $w_1 * w_2 * w_3 * w_4$, in such a way that the same crossing sequence occurred between any two consecutive parts. Then, by excluding w_2 if the maximum number of crossings is not attained while M scans w_2, and excluding w_3 otherwise, one would obtain an input z (i.e. $w_1 * w_3 * w_4$ or $w_1 * w_2 * w_4$, respectively) shorter than x_n and such that $cr(z) = cr(x_n)$, contradicting the choice of x_n.

This shows that the number of distinct crossing sequences of M on cells of the input x_n is at least $\frac{|x_n|-1}{2}$.

Since a crossing sequence is a sequence of states (with possible repetitions) and M has m states, for any input there can be at most m^l distinct crossing sequences of length $\leq l$, i.e. at most l distinct crossing sequences of length $\leq \log_m l = \frac{\log l}{\log m}$. In particular, there can be at most $\frac{|x_n|+1}{4}$ distinct crossing sequences of length $\leq \frac{1}{\log m} \cdot \log \frac{|x_n|+1}{4}$ on x_n.

Since we proved above that the number of distinct crossing sequences of M on x_n is at least $\frac{|x_n|-1}{2}$, this leaves at least $\frac{|x_n|-1}{4}$ distinct crossing sequences of length $\geq \frac{1}{\log m} \cdot \log \frac{|x_n|+1}{4}$ on x_n. Hence

$$t(x_n) \geq \frac{|x_n|-1}{4} \cdot \frac{1}{\log m} \cdot \log \frac{|x_n|+1}{4}.$$

Since, for some constant $c > 0$ and for any sufficiently large $|x_n|$,

$$(|x_n|-1) \cdot \log \frac{|x_n|+1}{4} \geq \frac{1}{c} \cdot |x_n| \cdot \log |x_n|,$$

we have for any sufficiently large $|x_n|$

$$t(x_n) \geq \frac{1}{4c \log m} \cdot |x_n| \cdot \log |x_n|,$$

and hence M takes at least time proportional to $|x| \cdot \log |x|$ for infinitely many x. $\square$

Thus TIMEF $[\,|x|{\cdot}\log |x|\,]$ is the next time complexity class after TIMEF $[\,|x|\,]$. Hartmanis [1968] has shown that there are indeed (0,1-valued) functions computable in time $|x| \cdot \log |x|$ but not in time $|x|$, and hence the result cannot be improved.

Constant space

From the Union Theorem VII.3.7 we know that there is a gap at the lower end of the space complexity classes, consisting of the *functions computable in constant space*. The class of sets (i.e. 0,1-valued functions) computable in constant space deserves a special name, because of its interest.

Definition VIII.1.4 CONSPACE *is the class of sets computable in deterministic constant space.*

NCONSPACE *is the class of sets acceptable in nondeterministic constant space.*

CONSPACE is a widely studied class which admits a number of equivalent characterizations, some of which we now consider (for further information see p. I.52 and Eilenberg [1974], [1976], Hopcroft and Ullman [1979], Lewis and Papadimitriou [1981], Wagner and Wechsung [1986], Perrin [1990]).

First, suppose that a set A is computed in constant space c by a Turing machine. By first eliminating the output activity (which consists only of writing a 0 or a 1) in favor of special halting states (respectively rejecting or accepting the input), then by enlarging the number of symbols of the alphabet (to step from space c to space 1) and the number of states (to step from space 1 to space 0), one can restrict attention to Turing machines that only read the input, and either accept it or reject it, without ever writing. These machines are called **two-way finite automata**, and one can take as an equivalent definition of computability of A in constant space, the existence of a two-way finite automaton M that *computes* A, in the sense that it accepts inputs in A and it rejects inputs not in A. In particular, this means that *constant space is equivalent to no space at all*.

If one imposes a further restriction on a two-way finite automaton by requiring the input head to move in only one direction (and, by standard convention, having the head positioned at the beginning on the leftmost cell of the input), one obtains the class of **finite automata** (McCulloch and Pitts [1943]).

One can also define **nondeterministic (two-way) finite automata** (Rabin and Scott [1959]) in the obvious way, by admitting (two-way) automata whose instructions are not necessarily consistent, i.e. such that more than one is possibly applicable in a given situation. We then say that an input is accepted if there is at least one computation (i.e. one behavior of the machine) that accepts it, and is rejected otherwise (i.e. if every halting computation rejects it). If M accepts inputs in A and rejects inputs not in A, then, in accordance with VII.4.18, we say that M *accepts* A.

The next result shows that the class of sets computed or accepted by the four models of finite automata just introduced is the same, and thus has a certain stability.

Theorem VIII.1.5 Machine Characterizations of CONSPACE (Rabin and Scott [1959], Shepherdson [1959]) CONSPACE *is the class of sets:*

1. *computed by finite automata*
2. *accepted by nondeterministic finite automata*
3. *computed by two-way finite automata*
4. *accepted by nondeterministic two-way finite automata.*

Proof. Before turning to the equivalence proof, we briefly discuss how automata can be described. Recall that a Turing machine is completely determined by a set of instructions of the form

$$q_a s_b s_c q_d j,$$

which state that when the machine is in state q_a and reads s_b, it must print s_c in place of s_b, change its state to q_d and (depending on whether $j = 0, 1, 2$) stay still, move one cell to the right, or move one cell to the left.

For *automata*, since no writing activity is involved, we can omit in the instructions any mention of s_c.

For *two-way automata*, we can also suppose that j is either 1 or 2. Indeed, if the head does not move, then it still reads the same symbol (because there is no writing activity), and thus it simply performs a sequence of state changes before moving. Then a single instruction can tell the machine to step directly to the first state that forces a movement, and one can systematically replace all sequences of $n+1$ instructions, of which the first n do not make the head move, by a single instruction. Notice that it is thus enough to restrict attention to a number n bounded by the number of states, since terminating computations (in the nondeterministic case: of minimal length) cannot repeat the same state twice without moving.

For *one-way automata*, we can actually drop the part relative to j, by adopting the convention that the head (starting on the leftmost cell of the input) always moves right, unless it reaches a final state (which it does, at the latest, when it steps out of the input).

We can now turn to the equivalence proof, which consists of three parts.

- *reduction of deterministic two-way finite automata to nondeterministic finite automata*

  Suppose a deterministic two-way automaton M computes a set A. The computation on input x can be completely described by the crossing sequences (see p. 73) of M. The idea is to simulate the two-way computation of M by a one-way computation consisting of steps from a crossing sequence and a scanned symbol to the next crossing sequence. Thus we build a finite automaton whose states are crossing sequences of M.

  Since there is no writing and there can be no loop (because M always halts, and either accepts or rejects), no crossing sequence may have a repeated state with the head going in the same direction. Since two subsequent states in a crossing sequence must correspond to different directions of the head movement (precisely, the odd ones correspond to left-to-right, and the even ones to right-to-left), there are only finitely many possible crossing sequences of M, namely those of length at most twice the number of states, and with no state repeated in the even or odd places.

  Given two such crossing sequences $c = \langle q_1, q_2, \ldots\rangle$ and $c' = \langle q_1', q_2', \ldots\rangle$ and a symbol a, one can check whether c and c' are two crossing sequences matching w.r.t. a (in the sense that they can be part of a computation of M, respectively at the left and at the right of a), by the following recursive procedure.

  Since the first state q_1 in c corresponds to a movement from left to right, M finds itself in state q_1 reading a, and we can check whether there is an instruction telling M in state q_1 and reading a to do one of the following two things:

  1. change its state to q_2 and move one cell to the left (this corresponds to a reversal in the head direction)
  2. change its state to q_1' and move one cell to the right (this corresponds to a move in the same direction as the previous one).

  In case 1, one starts the procedure again, with q_3 (which is a state corresponding to a movement from left to right) in place of q_1, and continues until a state q_{2i+1} is found such that case 2 applies. In case 2, since q_2' in c' corresponds to a movement from right to left, M finds itself in state q_2' reading a, and we can check whether there is an instruction telling M in state q_2' and reading a to do one of the following two things:

3. change its state to q'_3 and move one cell to the right
4. change its state to q_{2i+2} and move one cell to the left.

In case 3, one starts the procedure again, with q'_4 in place of q'_2, and continues until a state q'_{2j+2} is found such that case 4 applies. In case 4, one goes back to the beginning, with q_{2i+3} in place of q_1 and q'_{2j+3} in place of q'_1.

Thus c and c' are eventually matched as follows:

$$q_1, \ldots, q_{2i+1}, q'_1, \ldots, q'_{2j+2}, q_{2i+2}, \ldots, q'_{2j+3}, \ldots$$

The problem is that, given a crossing sequence c and a symbol a, there may be more than one crossing sequence c' as above. We thus actually build a *nondeterministic* finite automaton N, as follows:

- the alphabet of N is the same as the alphabet of M
- the states of N are possible crossing sequences of M (in particular, N can have exponentially more states than M)
- the initial states of N are states beginning with the initial state of M
- the halting states of N are states ending in a halting state of M
- the instructions of N are all triples cac' where c and c' are states of N matching w.r.t. a.

Since any computation of M on a given input produces a set of matching crossing sequences (w.r.t. subsequent symbols of the input), and hence a possible computation of N on the same input, the set computed by M is a subset of the set accepted by N.

Conversely, since any accepting computation of N on a given input corresponds to an accepting computation of M on the same input (because M is deterministic, and hence two crossing sequences match in at most one way), the set accepted by N is a subset of the set computed by M.

Thus the set computed by M is equal to the set accepted by N.

- *reduction of nondeterministic two-way finite automata to nondeterministic finite automata*
  In the case of a nondeterministic two-way automata M, it is no longer true that if M accepts a set A, then a crossing sequence cannot have a repeated state with the head going in the same direction. Indeed, inconsistent instructions provide a choice, and a repeated state does not forbid the possibility of continuing the computation in a different way

after finitely many repetitions of the same behavior. Since there may be infinitely many crossing sequences, it is not possible to use *all* of them as states of a finite automaton.

However, among the possible halting computations on a given input, there are minimal ones in which no crossing sequence has a repeated state with the head going in the same direction. What happens between any two such states is irrelevant for the outcome of the computation, and is avoided by the computation that makes the appropriate final move the first time such a state occurs.

By restricting attention to crossing sequences of length at most twice the number of states and with no state repeated in the even or odd places, one thus restricts the set of acceptable computations without missing any of their outcomes. The proof can then proceed as in the case of deterministic two-way automata.

- *reduction of nondeterministic finite automata to deterministic finite automata*
  Suppose a nondeterministic finite automaton N accepts a set A. Each possible computation on input x can be completely described by a sequence of states of length at most $|x|$, and thus the set of all possible computations can be described by a tree of height at most $|x|$, whose branches describe all possible computations. This tree is finitely branching, since a bound on the number of possible successors of a given node is given by the number of instructions of N. Recall that an input is accepted if there is at least one branch ending in an accepting halting state, while it is rejected if all branches end in a rejecting halting state.

  The idea is to simulate the tree corresponding to all possible nondeterministic computations of N on a given input by a single deterministic computation consisting of steps from one level of the tree and a scanned symbol to the next level of the tree. Since there may be infinitely many different levels (for computations trees on different inputs), it is not possible to use *all* of them as states of a finite automaton.

  However, the number of states of N occurring at any given level is finite, and we can thus use not the levels themselves, but the sets of states occurring at such levels. We thus build a deterministic finite automaton M, as follows:

  – the alphabet of M is the same as the alphabet of N

  – the states of M are all subsets of the set of states of N (in particular, M can have exponentially more states than N)

– the initial state of M is the singleton consisting of the initial state of N
– the halting states of M are $\emptyset$ (rejecting) and the states of M containing a halting accepting state of N (accepting)
– the instructions of M are all triples QaQ' where

$$Q' = \{q' : (\exists q)(q \in Q \ \wedge \ qaq' \text{ is an instruction of } M)\}.$$

Since any accepting computation of N produces (a sequence of states of N ending in a halting accepting state, and hence) an accepting computation of M, and any rejecting computation of N (forces every sequence of states of N to end in a halting rejecting state, and hence) produces a rejecting computation of M, the set accepted by N is contained in the set computed by M.

Conversely, since every computation of M corresponds to a tree of possible computations of N, and is accepting if and only if one branch ends in an accepting state of N, the set computed by M is contained in the set accepted by N.

Thus the set accepted by N is equal to the set computed by M. □

Corollary VIII.1.6 CONSPACE = NCONSPACE.

Chandra and Stockmeyer [1976] have supplemented the previous result, by proving that CONSPACE is also the class of sets accepted by two-way *alternating finite automata*.

CONSPACE is defined in terms of constant *space* computations. The next proposition characterizes it in terms of *time* bounded computations, and it refines a special case of the Space-Time Trade-Off Theorem VII.4.17.

Proposition VIII.1.7 (Trakhtenbrot [1964], Hennie [1965]) *A set is computable in constant space if and only if it is computable in real time.*

Proof. If A is computable in constant space, then A is computable by a finite automaton by VIII.1.5, and hence one knows whether $x \in A$ at the latest after having scanned the input, i.e. after $|x|$ moves.

Alternatively, if A is computable in constant space, the proof of VII.4.17.2 (see VII.4.5) shows that A is also computable in time $c \cdot |x|$ for some constant c. We cannot directly appeal to the Time Linear Speed-Up Theorem VII.4.8, because the function $|x|$ is not large enough to satisfy its hypothesis. However, if A were not computable in time $|x|$, then by VIII.1.3 any Turing machine computing A would take at least $|x| \cdot \log |x|$ steps for infinitely many x, while $c \cdot |x| < |x| \cdot \log |x|$ for almost every x.

For the converse, let A be computable in real time by a Turing machine M. The proof of VIII.1.3 shows that the function $cr(x)$ (defined there as the maximum number of crossings between two adjacent cells of the input x during the computation of M) is bounded by a constant c (otherwise A would take at least $|x| \cdot \log |x|$ steps for infinitely many x). Then A is acceptable in nondeterministic constant space (and hence, by VIII.1.5, computable in deterministic constant space), by the machine N defined as follows.

On input x, N starts on the rightmost cell of the input, guesses a crossing sequence of M of length $\leq c$ (which can be written in constant space for all x), and checks whether the sequence is compatible with the work of M on the right of the input. This can be done, since the behavior of M on the right of the input depends only on the crossing sequence (as the tape is empty at the beginning, except for the input).

If the check fails, N rejects x. Otherwise, N moves one cell to the left, guesses another crossing sequence, and again checks whether this is a crossing sequence of M compatible with the work of M to the right of the scanned cell. This can be done, since the behavior of M on the right of the scanned cell depends only on the symbol read and the previous crossing sequence.

If the check fails, N rejects x. Otherwise, N continues the procedure until it gets to the end of the input x. After the last crossing sequence has been guessed and checked, one knows whether the sequence of guessed crossing sequences corresponded to an accepting or rejecting computation of M on x, and thus N simulates M nondeterministically in constant space (since at every step at most two crossing sequences need to be considered, each of which can be written in constant space). $\Box$

Corollary VIII.1.8 CONSPACE = REAL.

Exercise VIII.1.9 Regular sets. The class of **regular sets** of strings of 0's and 1's is defined inductively as the smallest class containing $\emptyset$, $\{0\}$ and $\{1\}$, and closed under set-theoretical union $A \cup B$, product $A \cdot B$ (where $A \cdot B$ is the set of strings obtained by concatenating one string of A and one of B), and Kleene's closure A^* (where A^* is the set of strings obtained by concatenating any finite number of strings of A) (Kleene [1956]).

CONSPACE *is the class of regular sets.* (Kleene [1956]) (Hint: one direction follows by induction on the definition of regular sets. For the converse, note that the only important things in a computation by a finite automaton are the scanned symbols and the states. Thus the behavior can be pictured by a directed graph, whose vertices are the states and whose edges (indexed by 0's and 1's) correspond to state transitions determined by the scanned symbols. The accepted strings correspond to paths through the graph, going from the initial state to any accepting halting state. Let $P^k_{i,j}$ be the set of paths going from q_i to q_j through $\{q_1, \ldots, q_k\}$. Obviously, if $k > 1$,

$$P^k_{i,j} = P^{k-1}_{i,j} \cup P^{k-1}_{i,k} \cdot (P^{k-1}_{k,k})^* \cdot P^{k-1}_{k,j},$$

where $*$ indicates loops and $\cdot$ means concatenation of paths. Therefore, by induction on k, it follows that for all $k > 1$ the set of strings corresponding to paths in $P^k_{i,j}$ is regular. This completes the proof, because a string is accepted by an automaton M with states $q_1, \ldots, q_n$ if and only if it corresponds to a path in

$$\bigcup \{P^n_{1,j} : q_j \text{ is an accepting halting state}\},$$

i.e. if and only if it is in the union of finitely many regular sets.)

Exercise VIII.1.10 Regular grammars. Recall, from Section II.1, that a **grammar** consists of a finite alphabet (possibly with distinguished symbols), a finite set of axioms and a finite set of finitary rules (called *productions*) of the form $\sigma \to \tau$ stating that if a string σ occurs in a word, then it can be replaced by the string τ (Thue [1914]). A **regular grammar** is a grammar whose productions are either all of the form

$$x \to \sigma y \qquad x \to \sigma,$$

or all of the form

$$x \to y\sigma \qquad x \to \sigma,$$

where x and y stand for individual symbols and σ for (possibly empty) strings (Chomsky [1956]).

CONSPACE *is the class of sets generated by regular grammars.* (Chomsky and Miller [1958]) (Hint: productions of a regular grammar and transition functions of a nondeterministic finite automaton are easily put into correspondence.)

We now look beyond constant space. The next proposition, which is an analogue of VIII.1.3, shows that nothing is gained unless at least $\log\log|x|$ space is allowed.

Proposition VIII.1.11 (Hartmanis, Lewis and Stearns [1965]) *If a function is not computable in constant space, then any Turing machine computing it must take at least space proportional to* $\log\log|x|$ *for infinitely many* x.

Proof. Given f not computable in constant space, let M be a Turing machine (with an alphabet of p symbols and m states) computing f in space $s(x)$. By the proof of VII.4.5, there is only a finite number $c_x \le m \cdot s(x) \cdot p^{s(x)}$ of possible configurations of the working tape on input x. We enumerate these configurations in some order, and refer to them by their position in the enumeration.

Given an input $x = w * z$, let the **transition matrix**

$$T_{x,w} = [t_{i,j}]_{1 \le i,j \le c_x}$$

be defined as follows. If the input head of M is on the leftmost symbol of z, then $t_{i,j}$ is the (uniquely determined) two-digit number defined as follows:

- If M in configuration i reaches configuration j without leaving z, the first digit of $t_{i,j}$ is 1. Otherwise, the first digit of $t_{i,j}$ is 0.

- If M in configuration i eventually leaves z, and reaches configuration j immediately after the crossing from z to w, the second digit of $t_{i,j}$ is 1. Otherwise, the second digit of $t_{i,j}$ is 0.

Since f is not computable in constant space, $s(x)$ cannot be bounded by a constant. Since $s(x)$ is unbounded, given n there is an input x_n of smallest length such that $s(x_n) \geq n$.

The computation of the lower bound in the statement of the proposition rests on the following fact:

- *no transition matrix can occur twice on different beginnings of x_n*
  Otherwise, one could split the input x_n into three parts $w_1 * w_2 * w_3$ with w_2 nonempty, in such a way that the same transition matrix occurred for the different beginnings w_1 and $w_1 * w_2$ (i.e. $T_{x_n,w_1} = T_{x_n,w_1*w_2}$). Then, by excluding w_2, one would obtain an input $w_1 * w_3$ shorter than x_n. To reach a contradiction with the choice of x_n, it is enough to show that $s(w_1 * w_3) \geq n$.

  Consider the computation of M on input x_n. Since $s(x_n) \geq n$ by definition of x_n, there is a configuration of the working tape that uses at least n cells. If such a configuration is reached while the input head (starting with the leftmost cell of the input) does not leave w_1, then it is reached on any input starting with w_1, in particular on $w_1 * w_3$.

  Otherwise, the input head eventually crosses the boundary between w_1 and $w_2 * w_3$ for the first time from left to right, which results in the input head scanning the leftmost cell of $w_2 * w_3$. Let i be the configuration of the working tape at that instant. If M reaches a configuration j that uses at least n cells without leaving $w_2 * w_3$, the first digit of $t_{i,j}$ in T_{x_n,w_1} is 1. Since $T_{x_n,w_1} = T_{x_n,w_1*w_2}$, the first digit of $t_{i,j}$ is also 1 in T_{x_n,w_1*w_2}, i.e. M also reaches configuration j without leaving w_3, if the input head is on the leftmost cell of w_3 and the configuration is i. Then M does so on input $w_1 * w_3$, since i is the configuration that M finds itself in immediately after leaving w_1 (independently of the rest of the input).

  If M does not reach a configuration that uses at least n cells without leaving $w_2 * w_3$, the input head must again cross the boundary between w_1 and $w_2 * w_3$, this time from right to left. Let j be the configuration reached immediately after the crossing. Then the second digit of $t_{i,j}$ in T_{x_n,w_1} is 1. Since $T_{x_n,w_1} = T_{x_n,w_1*w_2}$, the second digit of $t_{i,j}$ is also 1 in T_{x_n,w_1*w_2}, i.e. M also crosses the boundary on the left of w_3 and finds itself in configuration j immediately after the crossing, if the input head is on the leftmost cell of w_3 and the configuration is i. Then M does so on input $w_1 * w_3$, since i is the configuration that M finds itself in immediately after leaving w_1 (independently of the rest of the input).

Now the input head is on w_1 once more, and one can start again, i.e. a configuration that uses at least n cells is reached by either no longer leaving w_1, or by crossing the boundary between w_1 and $w_2 * w_3$ but no longer leaving $w_2 * w_3$, or by crossing the boundary again. These three cases are treated as above, and after a finite number of steps, the first configuration that uses at least n cells is reached on input $w_1 * w_2 * w_3$, and hence on input $w_1 * w_3$.

This shows that the number of distinct $c_{x_n} \times c_{x_n}$ transition matrices of M on input x_n is at least $|x_n| - 1$.

Since the number of $c_{x_n} \times c_{x_n}$ transition matrices is $4^{c_{x_n}^2}$ (because each $t_{i,j}$ can take one of four different values) and $c_{x_n} \leq m \cdot s(x_n) \cdot p^{s(x_n)}$,

$$|x_n| - 1 \leq 4^{[m \cdot s(x_n) \cdot p^{s(x_n)}]^2}.$$

By taking logarithms twice,

$$\begin{aligned} \log\log(|x_n| - 1) &\leq 1 + 2 \cdot [\log m + \log s(x_n) + s(x_n) \cdot \log p] \\ &\leq c \cdot s(x_n), \end{aligned}$$

for some constant $c > 0$. Since, for any sufficiently large $|x_n|$,

$$\frac{1}{2} \cdot \log\log|x_n| \leq \log\log(|x_n| - 1),$$

we have that for any sufficiently large $|x_n|$

$$\frac{1}{2c} \cdot \log\log|x_n| \leq s(x_n),$$

and hence M takes at least space proportional to $\log\log|x|$ for infinitely many x. $\square$

Thus SPACEF$[\log\log|x|]$ is the next space complexity class after CONSPACE. Hartmanis, Lewis and Stearns [1965] have shown that there are (0,1-valued) indeed functions computable in space $\log\log|x|$ but not in constant space, and hence the result cannot be improved.

Notice that the gap just found for space is much smaller than the gap found in VIII.1.3 for time. In particular, it allows the consideration of a natural space bound between constant space and space $|x|$, namely $\log|x|$ (which, by VIII.1.1, does not instead produce a new time bound).

Exercise VIII.1.12 *If a set is not acceptable in nondeterministic constant space, then every Turing machine accepting it must take at least* $\log\log|x|$ *nondeterministic space for infinitely many* x. (Hopfcroft and Ullman [1969]) (Hint: the same proof of VIII.1.11 works if one modifies the definition of a transition matrix by considering whether certain configurations *can* be reached by a possible computation of the machine.)

Logarithmic space

From VII.4.13, VII.4.17.2, VII.4.20, VII.4.23, VII.4.24 and VII.4.25.2 we know that a logarithmic lower bound is a minimal condition for results on space to go through. We now investigate the class of sets obtained by imposing precisely such a bound.

VIII.1.6 shows that the same sets are computable in deterministic and nondeterministic constant space. While VII.4.23 shows that the same is true of most space complexity classes, it leaves open precisely the case of logarithmic space. For this reason, we introduce two separate classes.

Definition VIII.1.13 LOGSPACE *is the class of sets computable in deterministic logarithmic space in the length of the input.*

NLOGSPACE *is the class of sets acceptable in nondeterministic logarithmic space in the length of the input.*[1]

Complementing similar machine-theoretical characterizations of CONSPACE (see VIII.1.5), Hartmanis and Hunt [1975] have proved that LOGSPACE is the class of sets computed by deterministic two-way *multihead finite automata*, and NLOGSPACE is the class of sets accepted by nondeterministic two-way multihead finite automata.

Since nondeterministic machines include the deterministic ones, we obviously have

$$\text{LOGSPACE} \subseteq \text{NLOGSPACE},$$

but it is not known whether the inclusion is proper. Monien [1975] has shown that LOGSPACE = NLOGSPACE if and only if LOGSPACE is closed under Kleene's closure operation $*$ defined in VIII.1.9.

One possible attempt to prove that NLOGSPACE and LOGSPACE are distinct would be to exploit the analogy between corresponding deterministic and nondeterministic complexity classes on the one hand, and the recursive and r.e. set on the other hand, by trying to adapt the proof that there is an r.e. nonrecursive set (II.2.3). A general discussion of what goes wrong in such attempts is given in Section 3, for the more important case of NP and P.

[1]In the literature, LOGSPACE and NLOGSPACE are sometimes called L and NL.

As for the analogy between NLOGSPACE and the class of the r.e. sets, the next result shows that it is not perfect, since the r.e. sets are not closed under complement (see II.1.19).

Proposition VIII.1.14 (Immermann [1988], Szelepcsényi [1988]) NLOGSPACE *is closed under complement, i.e. a set A is in* NLOGSPACE *if and only if its complement is.*

Proof. By VII.4.23. □

However, it is possible to introduce an analogue of m-reducibility (III.2.1) and prove the existence of an analogue of m-complete r.e. sets (III.2.4).

Definition VIII.1.15 (Stockmeyer and Meyer [1973], Jones [1975]) *A is* **logarithmic space m-reducible** *to B ($A \leq_m^{\text{LOGSPACE}} B$) if, for some function f computable in logarithmic space in the length of the input,*

$$x \in A \;\Leftrightarrow\; f(x) \in B.$$

Exercises VIII.1.16 (Stockmeyer and Meyer [1973], Jones [1975])

a) $\leq_m^{\text{LOGSPACE}}$ *is reflexive and transitive.* (Hint: transitivity is not trivial, the problem being that to compute $f(g(x))$ we would like to write $g(x)$ on the working tape and then simulate a computation of f on $g(x)$. But if g is computable in logarithmic space, by VII.4.17 we only know that it is computable in time $|x|^a$ for some a, and hence $|g(x)| \leq |x|^a$. Thus it could be that $|g(x)| > \log|x|$, and to record $g(x)$ would take us out of the allowed working space. To get around this problem, note that to simulate f on $g(x)$ we just need to know, at any stage, the symbol that the input head would scan. Since $|g(x)| \leq |x|^a$, we can record the *length* of the input $g(x)$ in $a \cdot \log|x|$ space, on one working tape. When at a certain point the machine computing f is scanning the i-th symbol of the input and moves right or left, we simulate g long enough to produce the $(i+1)$-th or $(i-1)$-th symbol of the input, and this takes only $\log|x|$ space, since g is computable in logarithmic space. By VII.4.2, the output tape of our model of a Turing machine is write-only, and thus the behavior of the Turing machine does not depend on what is written on it. We can thus simulate $f(g(x))$ in space proportional to $\log|x|$, and hence in space $\log|x|$ by the Space Linear Speed-Up Theorem VII.4.8.)

b) *If A is in* LOGSPACE, *then $A \leq_m^{\text{LOGSPACE}} B$ for any set $B \neq \emptyset, \omega$.*

c) *If $A \leq_m^{\text{LOGSPACE}} B$ and B is in* LOGSPACE, *then so is A.*

d) *If $A \leq_m^{\text{LOGSPACE}} B$ and B is in* NLOGSPACE, *then so is A.* (Hint: from part a).)

The next definition is patterned on III.2.5.

Definition VIII.1.17 *Given a class $\mathcal{C}$ of sets, a set A is* **logarithmic space m-complete**, *or* **log-space complete**, *for $\mathcal{C}$ if it is in $\mathcal{C}$, and $B \leq_m^{\text{LOGSPACE}} A$ for every set B in $\mathcal{C}$.*

Theorem VIII.1.18 (Savitch [1970]) *There are* log*-space complete sets for* NLOGSPACE*.*

Proof. Recall that a straightforward way of obtaining complete sets for the class of r.e. sets was by coding them into a master set defined (in II.2.7) by

$$\langle x, e\rangle \in \mathcal{K}_0 \;\Leftrightarrow\; M_e \text{ halts on input } x.$$

Here we proceed similarly, from a canonical enumeration $\{N_e\}_{e\in\omega}$ of the nondeterministic Turing machines, and define

$$\langle x, e, n\rangle \in \mathcal{K}_0^{\text{NLOGSPACE}} \;\Leftrightarrow\; N_e \text{ accepts } x \text{ in } \log|n| \text{ space.}$$

- $\mathcal{K}_0^{\text{NLOGSPACE}} \in$ NLOGSPACE
  Given an input $\langle x, e, n\rangle$ we decode it into x, e and n, mark a space $\log|n|$ on the working tape (which is possible, because $\log|n|$ is a space constructible function), and simulate N_e on input x and within space $\log n$. We accept $\langle x, e, n\rangle$ if the simulation of N_e accepts x, and reject it if the simulation rejects, or tries to step out of the marked space, or enters a loop.

  As in the proof of VII.4.13, we can detect whether a loop occurs in $\log|n|$ space, and thus the whole procedure requires only log-space, if the *decoding procedure* works in log-space and is such that $|n| \le |\langle x, e, n\rangle|$, since then $\log|n| \le \log|\langle x, e, n\rangle|$.

- *if* $B \in$ NLOGSPACE*, then* $B \le_m^{\text{LOGSPACE}} \mathcal{K}_0^{\text{NLOGSPACE}}$
  Let N_e be a nondeterministic Turing machine accepting B in space $\log|x|$, and let a be the constant loss needed to simulate N_e on a fixed Turing machine N for $\mathcal{K}_0^{\text{NLOGSPACE}}$. Then

$$\begin{aligned} x \in B \quad &\Leftrightarrow \quad N_e \text{ accepts } x \text{ in space } \log|x| \\ &\Leftrightarrow \quad N \text{ accepts } x \text{ in space } a\log|x| = \log|x|^a \\ &\Leftrightarrow \quad \langle x, e, |x|^a\rangle \in \mathcal{K}_0^{\text{NLOGSPACE}}. \end{aligned}$$

  The function $f(x) = \langle x, e, |x|^a\rangle$ is log-space computable if the *coding procedure* is log-space computable, and then from

$$x \in B \;\Leftrightarrow\; f(x) \in \mathcal{K}_0^{\text{NLOGSPACE}}$$

  we have $B \le_m^{\text{LOGSPACE}} \mathcal{K}_0^{\text{NLOGSPACE}}$.

It remains only to define coding and decoding procedures working in log-space. This can easily be done for strings of any length, by letting $\langle x_1, \ldots, x_n\rangle$ be the string obtained from $x_1, \ldots, x_n$ by the following procedure:

- substitute 10 and 11 for 0 and 1, respectively, for each digit of the binary representation of x_i
- concatenate the new strings thus obtained, with a 00 inserted between them as a separation mark.

For example,

$$\langle 101, 1001\rangle = 1110110011101011.$$

Then both coding and decoding do not take any working space at all, since they can be obtained by direct transformations (one digit at a time in the coding direction, and two digits at a time in the decoding direction) of the input(s) into the output(s). Moreover, the conditions

$$|x_i| \leq |\langle x_1, \ldots, x_n\rangle|$$

are obviously satisfied, because

$$|\langle x_1, \ldots, x_n\rangle| = 2(\sum_{1\leq i\leq n} |x_i|) + 2(n-1). \quad \Box$$

Examples of real-life problems that turn out to be log-space complete for NLOGSPACE are given in Jones, Lien and Laaser [1976], Hopcroft and Ullman [1979], and Wagner and Wechsung [1986].

Obviously, every set in LOGSPACE is log-space complete for LOGSPACE. For nontrivial examples of sets complete for LOGSPACE w.r.t. stronger reducibilities, see Jones, Lien and Laaser [1976], and Cook and McKenzie [1987].

VIII.2 Deterministic Polynomial Time

We have considered the class of functions computable in real time $|x|$ in Section 1. What we said there, to the effect that this class is heavily dependent on the model of computation, also holds for any fixed polynomial bound $|x|^n$ (see Monien [1974] for a well-behaved model). Since, however, all the simulation results of Section VII.4 introduce only polynomial loss factors, the functions computable in time polynomial in $|x|$ form a minimal class with strong invariance properties, and thus a natural object of study.

Polynomial time computable functions

Definition VIII.2.1 (Von Neumann [1953], Cobham [1964], Edmonds [1965]) PF *is the class of functions computable in polynomial time in the length of the input.*

As already mentioned briefly above, by VII.4.6.2 and VII.4.7.b, the same class is obtained when considering n-tape ($n \geq 1$) or multitape Turing machines. Thus, **from this moment on, we drop the restrictions adopted in VII.4.1 and VII.4.2 on Turing machines** and use, for any particular result, the model that suits the proof better.

In terms of the notation of Chapter VII, the definition of PF can be restated as

$$\mathrm{PF} = \bigcup_{n \in \omega} \mathrm{TIMEF}\,[\,|x|^n\,].$$

While the classes $\mathrm{TIMEF}\,[\,|x|^n\,]$ are not independent of the model of computation, by VII.4.15 they do provide a hierarchy for PF.

Another hierarchy is provided by Kasai and Adachi [1980], in terms of a restriction of 'for' programs (defined in I.5.7).

Exercise VIII.2.2 $\{e : M_e$ *works in polynomial time* $\}$ *is* Σ^0_2*-complete.* (Hájek [1979]) (Hint: it is Σ^0_2 because

$$M_e \text{ works in polynomial time } \Leftrightarrow (\exists n)(\exists c)(\forall x)(\varphi_e(x)\downarrow \,\wedge\, T_e(x) \leq |x|^n + c).$$

To show Σ^0_2-completeness,define a recursive function such that

$$\mathcal{W}_e \text{ finite } \Leftrightarrow M_e \text{ works in polynomial time},$$

and use X.9.6.)

Closure properties

The notion of polynomial time computable function is defined in *complexity-theoretical* terms, by imposing a time bound on the Turing machine model of recursiveness. We now start an investigation of the effect of such a limitation on the other models of recursiveness introduced in Chapter I, with the goal of characterizing PF in *recursion-theoretical* terms.

We start with a simple but important observation.

Proposition VIII.2.3 *The following functions are polynomial time computable, together with their inverses:*

- *sum* $x + y$
- *product* $x \cdot y$
- *weak exponential* $x^{|y|}$

Proof. Almost all claims are trivial. For example, the usual algorithms for $x + y$ and $x \cdot y$ compute the values in roughly $\max\{|x|, |y|\}$ and $|x| \cdot |y|$ steps, and hence in time polynomially bounded in the length of the inputs.

Some care is needed for $x^{|y|}$, which is obtained by $|y|$ iterations of the product by x. At first sight, it would appear than $|y|$ iterations of a function computable in polynomial time $p(|\vec{x}|)$ would roughly require $(p(|\vec{x}|))^{|y|}$ steps, which is not a polynomial. The crucial observation is that, for all $z \leq |y|$, the values x^z are bounded in length by a fixed polynomial in $|x|$ and $|y|$, since

$$|x^z| \leq |(2^{|x|})^z| = |2^{|x|\cdot z}| \leq |x| \cdot z + 1 \leq |x| \cdot |y| + 1.$$

Then the time to compute each partial product is bounded by a fixed polynomial $p(|x|, |y|)$, and hence the time of the whole computation of $x^{|y|}$ is bounded by $|y| \cdot p(|x|, |y|)$, which is still a polynomial. $\Box$

Corollary VIII.2.4 *For every polynomial p, $2^{p(|x|)}$ is polynomial time computable.*

Proof. If

$$p(|x|) = a_n \cdot |x|^n + a_{n-1} \cdot |x|^{n-1} + \cdots + a_1 \cdot |x| + a_0$$

then

$$\begin{aligned}
2^{p(|x|)} &= 2^{a_n\cdot|x|^n + a_{n-1}\cdot|x|^{n-1} + \cdots + a_1\cdot|x| + a_0} \\
&= 2^{a_n\cdot|x|^n} \cdot 2^{a_{n-1}\cdot|x|^{n-1}} \cdots 2^{a_1\cdot|x|} \cdot 2^{a_0} \\
&= \underbrace{2^{|x|^n} \cdots 2^{|x|^n}}_{a_n \text{ times}} \cdot \underbrace{2^{|x|^{n-1}} \cdots 2^{|x|^{n-1}}}_{a_{n-1} \text{ times}} \cdots \underbrace{2^{|x|} \cdots 2^{|x|}}_{a_1 \text{ times}} \cdot 2^{a_0},
\end{aligned}$$

and

$$2^{|x|^m} = 2^{|x|\cdots|x|} = ((2^{|x|})^{|x|\cdots})^{|x|}.$$

That $2^{p(|x|)}$ is polynomial time computable follows from the fact that PF contains product and weak exponential, and is obviously under composition. $\Box$

While $2^{|x|}$ is polynomial time computable, the Time Hierarchy Theorem VII.4.15 implies that there are functions computable in time $2^{|x|}$ which are not polynomial time computable. It follows that PF *does not have the Ritchie-Cobham Property* VII.3.8 w.r.t. time, i.e. the implication

$$f \text{ is computable in time } t \in \mathrm{PF} \;\Rightarrow\; f \in \mathrm{PF}$$

does not always hold. To obtain a class with the Ritchie-Cobham Property w.r.t. time we have to go all the way up to the elementary functions (see VIII.7.6).

The proof that the weak exponential $x^{|y|}$ is polynomial time computable obviously fails for the usual exponential x^y, since it produces a polynomial in y and not in the length of y. In fact, we prove in VIII.2.17 that the exponential function is not polynomial time computable.

The trick of restricting attention to the weak exponential function is typical of the study of PF, in the following sense: most of the closure properties that hold for the primitive recursive functions and predicates w.r.t. operators bounded by the *size* of the inputs, also hold for the polynomial time computable functions and predicates w.r.t. operators bounded by the *length* of the inputs, as we now see.

Definition VIII.2.5 *A* **weak bounded operator** *is a bounded operator with a bound on the length of a number. In particular, we consider the following versions of the bounded operators defined in Section I.1:*

- *weak bounded sum* $\sum_{y \leq |z|}$
- *weak bounded product* $\prod_{y \leq |z|}$
- *weak bounded quantifiers* $\forall x \leq |y|$ *and* $\exists x \leq |y|$
- *weak bounded μ-recursion* $\mu y_{\leq |z|}$.

We now prove that weak operators preserve polynomial time computability.

Proposition VIII.2.6 Closure Properties of the Polynomial Time Computable Functions and Predicates (Cobham [1964])

1. *The polynomial time computable functions are closed under weak bounded sum, product and μ-recursion.*
2. *The polynomial time computable predicates are closed under logical connectives and weak bounded quantifiers.*

Proof. Closure under weak bounded sum and product is similar to the proof given above that the weak exponential is polynomial time computable (not surprisingly, since the weak exponential is itself a weak bounded product). For example, let

$$f(\vec{x}, z) = \prod_{y \leq |z|} g(\vec{x}, y)$$

and suppose $g(\vec{x}, y)$ is computable in time $p(|\vec{x}|, |y|)$, for some polynomial p. We first notice that, for all $y \leq z$, the values of $f(\vec{x}, y)$ are bounded in length by a fixed polynomial in $|\vec{x}|$ and $|y|$, since

$$|f(\vec{x}, y)| \leq |(\max_{y \leq |z|} g(\vec{x}, y))^{|z|}| \leq |z| \cdot p(|\vec{x}|, |z|),$$

where p is a polynomial bounding the time needed to compute g. Then the time needed to compute each partial product is bounded by a fixed polynomial $q(|\vec{x}|, |z|)$, and hence the time of the whole computation of $f(\vec{x}, z)$ is bounded by $|z| \cdot q(|\vec{x}|, |z|)$, which is still a polynomial.

Closure under logical connectives, weak bounded quantifiers and weak bounded μ-recursion then follows as on pp. I.25–26. In particular, the characteristic functions of $\neg P$, $P \wedge Q$ and $(\forall y \leq |z|)R(\vec{x}, y)$ are respectively $1 - c_P$, $c_P \cdot c_Q$ and $\prod_{y \leq |z|} c_R(\vec{x}, y)$. And

$$\mu y_{\leq |z|} R(\vec{x}, y) = \sum_{y \leq |z|} (y \cdot g(\vec{x}, y)),$$

where

$$g(\vec{x}, y) = \text{characteristic function of } R(\vec{x}, y) \wedge (\forall z < y) \neg R(\vec{x}, z). \quad \Box$$

Closure of the polynomial time computable predicates under the usual bounded quantifiers, i.e. of the form $\forall x \leq y$ and $\exists x \leq y$ rather than $\forall x \leq |y|$ and $\exists x \leq |y|$, is a major open problem. For a general discussion, see Section 4.

By the closure properties just proved, many of the numerical functions and predicates of common use proved to be primitive recursive in Section I.1 (see pp. I.24–26) turn out to be polynomial time computable, by the same proofs.

We now show that the same holds also for the numerical tools for arithmetization considered in Section I.7 (see pp. I.88–90).

We have already used polynomial time computable pairing functions in VIII.1.18, where $\langle x_1, \ldots, x_n \rangle$ was defined as the string obtained from $x_1, \ldots, x_n$ by the following procedure:

- substitute 10 and 11 for 0 and 1, respectively, for each digit of the binary representations of x_i
- concatenate the new strings thus obtained, with 00 inserted between them as a separation mark.

It is then immediate that the following functions and predicates are polynomial time computable (and, as already noted in VIII.1.18, computable without using any working space):

$$\begin{array}{rcl}
(x)_n & = & n\text{-th component of } x \\
ln(x) & = & \text{length of } x \\
Seq(x) & = & x \text{ is a sequence number} \\
x * y & = & \text{concatenation of } x \text{ and } y \\
x \sqsubseteq y & = & x \text{ is an initial subsequence of } y.
\end{array}$$

For example, to see whether x is a sequence number it is enough to see whether x is a sequence of two-digit blocks, none of which are of the form 01, and such that blocks of the form 11 are never consecutive, and do not occur at the beginning, nor at the end.

The final tool for arithmetization introduced in Section I.7 is course-of-value recursion (see I.7.1), which admits the following polynomial time version.

Proposition VIII.2.7 Weak Course-of-Values Recursion. *The class of polynomial time computable functions is closed under recursions in which the definition of $f(\vec{x}, y)$ may involve any number of (and possibly all) the values $\{f(\vec{x}, (y)_n)\}_{n \leq ln(y)}$ already obtained.*

Formally, let $\tilde{f}$ be the **weak history function** *of f, defined as:*

$$\tilde{f}(\vec{x}, y) = \begin{cases} \langle f(\vec{x}, (y)_1), \ldots, f(\vec{x}, (y)_{lny}) \rangle & \text{if } y \text{ is a sequence number} \\ 0 & \text{otherwise.} \end{cases}$$

Then, if f is defined as

$$f(\vec{x}, y) = \begin{cases} h(\vec{x}, y, \tilde{f}(\vec{x}, y)) & \text{if } y \text{ is a sequence number} \\ g(\vec{x}) & \text{otherwise} \end{cases}$$

and g, h are polynomial time computable, so is f.

Proof. The only interesting case is when y is a sequence number, since otherwise $f(\vec{x}, y)$ is equal to $g(\vec{x})$ and its computation is bounded in time by a polynomial in $\vec{x}$, by the hypothesis on g.

If y is a sequence number, then the computation of $f(\vec{x}, y)$ can be organized in the form of a tree of recursive calls. The basic observation is that the tree has both height and width bounded by $|y|$, because all the inputs needed for the computation of $f(\vec{x}, y)$ are hereditary components of the sequence number y. As in the proofs of VIII.2.3 and VIII.2.6 it follows that, for all hereditary components z of y, the values of $f(\vec{x}, z)$ are bounded in length by a fixed polynomial in $|\vec{x}|$ and $|y|$, and hence that the whole computation of $f(\vec{x}, y)$ is bounded in time by a polynomial in $\vec{x}$ and y. $\square$

We now have all the ingredients needed to strengthen the Normal Form Theorems I.7.3 and II.1.2 in a substantial way.

Theorem VIII.2.8 Improved Normal Form Theorem (Bereczki [1952], Grzegorczyck [1953], Smullyan [1961], Cobham [1964]) *There is a polynomial time computable function $\mathcal{U}$ and (for each $n \geq 1$) polynomial time computable predicates $\mathcal{T}_n$, such that for every partial recursive function φ of n variables there is a number e for which the following hold:*

1. $\varphi(x_1,\dots,x_n)\downarrow \Leftrightarrow \exists y\mathcal{T}_n(e,x_1,\dots,x_n,y)$

2. $\varphi(x_1,\dots,x_n) \simeq \mathcal{U}(\mu y\mathcal{T}_n(e,x_1,\dots,x_n,y))$.

Proof. We refer to the proof of II.1.2 and show that the primitive recursive functions and predicates in it can be systematically replaced by polynomial time computable functions and predicates. The real work for II.1.2 was done in the proof of I.7.3, using:

- the existence of a primitive recursive coding and decoding mechanism
- the closure of the class of primitive recursive predicates under logical connectives and bounded quantifiers
- the closure of the class of primitive recursive functions under course-of-value recursion.

Since we have already provided polynomial time versions of all these, it only remains to check that they suffice to obtain the result.

Indeed, a look at the predicates A, B, C, D and E used in the proof of I.7.3 shows that only quantifiers bounded by lengths are used there. Similarly, a look at the characteristic function of $\mathcal{T}$ used in the proof of I.7.3 shows that only weak course-of-value recursion is used there. □

Corollary VIII.2.9 *An n-ary relation P is r.e. if and only if there is an $n+1$-ary polynomial time computable relation R such that*

$$P(x_1,\dots,x_n) \Leftrightarrow \exists y R(x_1,\dots,x_n,y).$$

Proof. By II.1.10, we already know that a normal form for n-ary r.e. relations is

$$P(x_1,\dots,x_n) \Leftrightarrow \exists y\mathcal{T}_n(e,x_1,\dots,x_n,y) \quad \square$$

Exercises VIII.2.10 Polynomial time computable enumerating functions.

a) *Every nonempty r.e. set is the range of a polynomial time computable function.* (Kleene [1936], Rosser [1936], Bereczki [1952], Grzegorczyck [1953], Smullyan [1961], Skolem [1962], [1963]) (Hint: let A be the range of the recursive function f, and let $a \in A$. If

$$f(x) = \mathcal{U}(\mu y\, \mathcal{T}_1(e,x,y))$$

with $\mathcal{U}$ and $\mathcal{T}_1$ polynomial time computable, then A is the range of the polynomial time computable function

$$g(\langle x,y\rangle) = \begin{cases} \mathcal{U}(y) & \text{if } \mathcal{T}_1(e,x,y) \\ a & \text{otherwise.)} \end{cases}$$

b) *Not every infinite recursive set is the range of a one-one polynomial time computable function.* (Rosser [1936]) (Hint: let $\{f_e\}_{e\in\omega}$ be a recursive enumeration of the polynomial time computable functions. It is enough to put in $\overline{A}$ one element of the range of f_e, whenever f_e is one-one. We define A in stages, by putting one element into A at each stage. At stage n, consider the $n+1$ elements $f_n(0), \ldots, f_n(n)$. Either there is a repetition, and then f_n is not one-one, or one of them, say $f_n(z)$, is not yet in A. Put $f_n(z)$ in $\overline{A}$, and put in A the smallest element for which membership into A or $\overline{A}$ has not yet been determined. Then A is recursive, because membership for n is decided not later than stage n.)

Notice that *part b) holds for any r.e. class of total recursive functions*, e.g. for the primitive recursive functions, by the same proof.

Exercises VIII.2.11 Polynomial time computable graphs.

a) *A function has polynomial time computable graph if and only if it can be written in the form $\mu y R(x,y)$, with a polynomial time computable predicate R.* (Skolem [1944a]) (Hint: if f has polynomial time computable graph G_f, let $R = G_f$. Conversely, if R is polynomial time computable and $f(x) = \mu y R(x,y)$, then

$$G_f(x,y) \Leftrightarrow R(x,y) \wedge (\forall z < y)\neg R(x,z).)$$

b) *The functions with polynomial time computable graph are cofinal in the recursive functions*, i.e. given a recursive function f there is a function g with polynomial time computable graph such that g dominates f everywhere. (Axt [1959]) (Hint: $\mathcal{U}(x) < x$ by definition. If $f(x) = \mathcal{U}(\mu y \mathcal{T}_1(e,x,y))$ with $\mathcal{U}$ and $\mathcal{T}_1$ polynomial time computable, it is enough to let $g(x) = \mu y \mathcal{T}_1(e,x,y)$.)

c) *Every polynomial time computable function has polynomial time computable graph, but not conversely.* (Hint: if f is a recursive function whose graph is not polynomial time computable, then the function g defined in part b) is not polynomial time computable, since $\mathcal{U}$ is and $f = \mathcal{U} \circ g$.)

d) *Every function with polynomial time computable graph is recursive, but not conversely.* (Post [1946], Kleene [1947]) (Hint: the implication follows from part a). If f is a 0,1-valued recursive function which is not polynomial time computable, then $f(x) = \mu z_{\leq 1}(f(x) = z)$. If the graph of f were polynomial time computable, f would also be polynomial time computable.)

In contrast to part d), Skolem [1944] has shown that *every recursive function is the difference of two functions with polynomial time computable graph.*

The previous results show that even a small basis, such as the class of polynomial time computable functions and predicates, is sufficient to generate the whole classes of partial recursive functions and r.e. sets, together with the full power of the μ-operator.

Alternative characterizations

PF has been originally defined as the class of functions computable in polynomial *time* in the length of the input, i.e.

$$\mathrm{PF} = \{f : (\exists e)[f \simeq \varphi_e \ \wedge \ (\exists n)(\forall_\infty x)(T_e(x) \leq |x|^n)]\}.$$

Since the class of polynomials is closed under composition, PF can be more generally defined as the class of functions computable in polynomial *measure*, for any measure of complexity $\{\Phi_e\}_{e\in\omega}$ that is polynomially related to time on Turing machines, i.e.

$$\mathrm{PF} = \{f : (\exists e)[f \simeq \varphi_e \ \wedge \ (\exists n)(\forall_\infty x)(\Phi_e(x) \leq |x|^n)]\}.$$

Any such definition provides an alternative *complexity-theoretical* characterization of PF. However, any such characterization is still defined in terms of polynomial bounds on the amount of resource needed for the computation. We now look for alternative *number-theoretical* characterizations, that allow the recognition of a polynomial time computable function solely on the basis of its syntactic definition.

Since the length of each configuration of a Turing machine working in polynomial time is bounded by a polynomial, and the final configuration is obtained after a polynomial number of moves, the following operation is what one needs to be able to code the *global behavior* of a computation.

Definition VIII.2.12 *A function f is defined from g and h by* **polynomially bounded primitive recursion** *if there are polynomials p and q such that*

$$f(\vec{x}) = t(\vec{x}, p(|\vec{x}|)),$$

where

$$\begin{aligned} t(\vec{x}, 0) &= g(\vec{x}) \\ t(\vec{x}, y+1) &= h(\vec{x}, y, t(\vec{x}, y)) \end{aligned}$$

and

$$(\forall y \leq p(|\vec{x}|))[|t(\vec{x}, y)| \leq q(|\vec{x}|)].$$

Intuitively, polynomially bounded primitive recursion is a usual primitive recursion, with a polynomial *time* bound on the number of iterations and a polynomial *space* bound on the size of the intermediate values:

$$\underbrace{\boxed{\begin{array}{l} t(\vec{x},0) \\ t(\vec{x},1) \\ t(\vec{x},2) \\ \cdots \\ t(\vec{x},y) \end{array}}}_{q(|\vec{x}|)} \left.\vphantom{\begin{array}{l} t \\ t \\ t \\ \cdots \\ t \end{array}}\right\} p(|\vec{x}|)$$

To handle such a schema effectively, we need to be able to deal with *representations* of numbers, in addition to their *numerical values*. We thus extend the usual operations of successor and predecessor to numbers expressed in binary notation:

- *binary successors*

  $$\begin{aligned}\mathcal{S}_0(x) &= 2x\\ \mathcal{S}_1(x) &= 2x+1\end{aligned}$$

  $\mathcal{S}_0$ and $\mathcal{S}_1$ are the operations that correspond to adding, respectively, a 0 and a 1 at the end of the binary representation of x.

  Notice that $\mathcal{S}_1$ can be obtained from $\mathcal{S}_0$ and the usual successor $\mathcal{S}$, by composition.

- *binary predecessor*

  $$pd_0(x) \quad = \quad \text{(the integer part of) } \frac{x}{2}.$$

  pd_0 is the operation that chops off the last digit of the binary representation of a number.

The *local behavior* of a Turing machine, i.e. a single move, can easily be described by a nested definition by cases (describing the possible actions for any given combination of state and symbol). One thus arrives at the following characterization, which shows that PF is a polynomially bounded analogue of the class of primitive recursive functions.

Proposition VIII.2.13 (Buss [1986]) *The class of polynomial time computable functions is the smallest class of functions:*

1. *containing the initial functions, the binary successor and predecessor functions, and the characteristic function δ of equality*
2. *closed under composition*
3. *closed under definitions by cases*
4. *closed under polynomially bounded primitive recursion.*

Proof. Let $\mathcal{C}$ be the smallest class satisfying the stated conditions. The only nontrivial step in the proof that $\mathcal{C}$ contains only polynomial time computable functions is to show that PF is closed under polynomially bounded primitive recursion.

Given f defined from polynomial time computable functions g and h as in VIII.2.12, we show that f is polynomial time computable. Suppose T_g, T_h, T_t and T_f are the times needed to compute g, h, t and f. Then, roughly,

$$T_f(\vec{x}) = T_t(\vec{x}, 0) + \cdots + T_t(\vec{x}, p(|\vec{x}|)).$$

Since g and h are polynomial time computable by hypothesis, there are polynomials p_g and p_h such that, almost everywhere,

$$T_g(\vec{x}) \leq p_g(|\vec{x}|) \qquad \text{and} \qquad T_h(\vec{x}, y, z) \leq p_h(|\vec{x}|, |y|, |z|).$$

Then

$$T_t(\vec{x}, 0) \leq p_g(|\vec{x}|),$$

and

$$T_t(\vec{x}, y+1) \leq p_h(|\vec{x}|, |y|, |t(\vec{x}, y)|) \leq p_h(|\vec{x}|, p(|\vec{x}|), q(|\vec{x}|))$$

because, by hypothesis, $|t(\vec{x}, y)| \leq q(|\vec{x}|)$ and $|y| \leq y \leq p(|\vec{x}|)$. Then

$$T_f(\vec{x}) \leq p_g(|\vec{x}|) + p(|\vec{x}|) \cdot p_h(|\vec{x}|, p(|\vec{x}|), q(|\vec{x}|)),$$

and the right-hand-side is still a polynomial.

The idea to prove the converse, i.e. that every polynomial time computable function is in $\mathcal{C}$, has actually inspired the statement we are proving, and amounts to the following: given a Turing machine M computing a function f in polynomial time p, let

$$\begin{aligned} g(\vec{x}) &= \text{code of the initial configuration of } M \text{ on input } \vec{x} \\ h(\vec{x}, y, z) &= \text{code of the configuration of } M \text{ next to that coded by } z, \end{aligned}$$

where coding is done in the usual polynomial time way (see VIII.1.18). Polynomially bounded primitive recursion then produces

$$t(\vec{x}, p(|\vec{x}|)) = \text{code of the final configuration of } M \text{ on input } \vec{x},$$

because M works in time p. In particular, the maximum length of a configuration of M on input $\vec{x}$ is bounded by $p(|\vec{x}|)$, and by the properties of the coding mechanism, the length of the code of any configuration (i.e. $|t(\vec{x}, y)|$) is still bounded by a polynomial. Finally, $f(\vec{x})$ is then decoded from $t(\vec{x}, p(|\vec{x}|))$.

It remains to show that the coding and decoding mechanism can be defined in $\mathcal{C}$. First, one can easily deal with propositional connectives. For example,

$$c_{A \wedge B}(x) = \begin{cases} 0 & \text{if } c_A(x) = 0 \\ c_B(x) & \text{otherwise.} \end{cases}$$

Second, one can use definition by cases and polynomially bounded primitive recursion to define

$$\langle x_1, \ldots, x_n \rangle * \langle y_1, \ldots, y_m \rangle = \langle x_1, \ldots, x_n, y_1, \ldots, y_m \rangle$$

and

$$(\langle x_1, \ldots, x_n \rangle)_i = \begin{cases} n & \text{if } i = 0 \\ x_i & \text{if } 1 \leq i \leq n. \end{cases}$$

Note that the binary predecessor operation chops off the last digit of a binary string, and can thus be used for decoding. $\Box$

The previous result can still be seen as a characterization of PF based on polynomial bounds on a complexity measure, since polynomially bounded primitive recursion imposes an outside bound on the number of steps allowed to perform a primitive recursion, as well as on the length of the intermediate values. An intrinsically bounded primitive recursion is obtained by imposing a straight bound on the values of the function to be defined.

Definition VIII.2.14 *A function f is defined from functions g, h_0, h_1 and s by* **bounded primitive recursion on binary notation** *if*

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}) \\ f(\vec{x}, \mathcal{S}_0(y)) &= h_0(\vec{x}, y, f(\vec{x}, y)) \\ f(\vec{x}, \mathcal{S}_1(y)) &= h_1(\vec{x}, y, f(\vec{x}, y)), \end{aligned}$$

provided $y \neq 0$ in the second equation (otherwise $\mathcal{S}_0(y) = 0$) and

$$f(\vec{x}, y) \leq s(\vec{x}, y).$$

Intuitively, this is a usual primitive recursion done on binary notation, i.e. using $f(\vec{x}, y)$ to define $f(\vec{x}, y0)$ and $f(\vec{x}, y1)$, with an additional bound on the values:

$$\begin{array}{lcll} f(\vec{x}, 0) & = & g(\vec{x}) & \leq \; s(\vec{x}, 0) \\ f(\vec{x}, 1) & = & h_1(\vec{x}, 0, f(\vec{x}, 0)) & \leq \; s(\vec{x}, 1) \\ f(\vec{x}, 10) & = & h_0(\vec{x}, 1, f(\vec{x}, 1)) & \leq \; s(\vec{x}, 10) \\ f(\vec{x}, 11) & = & h_1(\vec{x}, 1, f(\vec{x}, 1)) & \leq \; s(\vec{x}, 11) \\ \ldots & & & \end{array}$$

Obviously, one could use n-ary notation for any $n \geq 2$ in a similar way.

Since we now have a bound on the values of the function f, to be able to obtain the polynomial time computable functions we only need a sufficiently large function to start with. The next result spells this out and shows that PF is a bounded analogue of the class of primitive recursive functions.

Proposition VIII.2.15 (Cobham [1964]) *The class of polynomial time computable functions is the smallest class of functions:*

1. *containing the initial functions, the binary successor functions and the weak exponential* $x^{|y|}$
2. *closed under composition*
3. *closed under bounded primitive recursion on binary notation.*

Proof. Let $\mathcal{C}$ be the smallest class satisfying the stated conditions. The only nontrivial step in the proof that $\mathcal{C}$ contains only polynomial time computable functions is to show that PF is closed under bounded primitive recursion on binary notation. By VIII.2.13, it is enough to reduce the latter to a polynomially bounded primitive recursion.

Let f be defined from polynomial time computable functions g, h_0, h_1 and s as in VIII.2.14. Notice that $f(\vec{x}, y)$ can be computed by an iteration of $|y|$ steps, by first computing the values at y, $pr_0(y)$, $pr_0(pr_0(y))$, $\ldots$ needed for its computation (this requires an iteration of the binary predecessor $|y|$ times), and then by iterating (again $|y|$ times) the functions h_0 and h_1 (according to whether the value considered at a given stage is even or odd). Since $f(\vec{x}, y) \leq s(\vec{x}, y)$ and s is polynomial time computable, the values of all steps in the iteration are bounded (in computation time, and hence) in length by a polynomial, as required by polynomially bounded primitive recursion.

Conversely, we use VIII.2.13 to prove that every polynomial time computable function is in $\mathcal{C}$. First, we notice that closure under bounded primitive recursion on binary notation and the presence of $x^{|y|}$ as a bound imply that $\mathcal{C}$ contains a number of polynomial time computable functions of common use, in particular $2^{p(|x|)}$ and other functions needed in the rest of the proof.

Second, we notice that closure of $\mathcal{C}$ under definition by cases follows from closure under composition and the presence of the following function:

$$C(x,y,z) = \begin{cases} x & \text{if } z = 0 \\ y & \text{otherwise.} \end{cases}$$

That C is in $\mathcal{C}$ follows from closure under bounded primitive recursion on binary notation, since

$$C(x,y,0) = x \qquad C(x,y,\mathcal{S}_0(z)) = C(x,y,\mathcal{S}_1(z)) = y \qquad C(x,y,z) \leq x + y.$$

It remains to prove closure of $\mathcal{C}$ under polynomially bounded primitive recursion. Let f be defined from functions g and h in $\mathcal{C}$ and polynomials p and q as in VIII.2.12, i.e.

$$f(\vec{x}) = t(\vec{x}, p(|\vec{x}|)),$$

where

$$\begin{aligned} t(\vec{x},0) &= g(\vec{x}) \\ t(\vec{x},y+1) &= h(\vec{x},y,t(\vec{x},y)) \end{aligned}$$

and

$$(\forall y \leq p(|\vec{x}|))[|t(\vec{x},y)| \leq q(|\vec{x}|)].$$

To show that $f \in \mathcal{C}$, we define

$$\begin{aligned} t_1(\vec{x},0) &= g(\vec{x}) \\ t_1(\vec{x},\mathcal{S}_0(y)) &= t_1(\vec{x},\mathcal{S}_1(y)) = \begin{cases} h(\vec{x},y,t_1(\vec{x},y)) & \text{if } |y| \leq p(|\vec{x}|) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

and notice that, if $|y| \leq p(|\vec{x}|)$,

$$t_1(\vec{x},y) = t(\vec{x},|y|-1).$$

The function t_1 is in $\mathcal{C}$ because is defined by bounded primitive recursion on binary notation, since

$$|t_1(\vec{x},y)| = |t(\vec{x},|y|-1)| \leq q(|\vec{x}|)$$

and hence

$$t_1(\vec{x},y) \leq 2^{q(|\vec{x}|)}.$$

It follows from

$$f(\vec{x}) = t(\vec{x},p(|\vec{x}|)) = t_1(\vec{x},2^{p(|\vec{x}|)})$$

that f is in $\mathcal{C}$ as well. $\Box$

The two characterizations of PF just given really amount to the same strategy, namely: to compute $f(\vec{x},y)$ we are allowed only polynomially many steps, each one with a polynomially bounded output length. While the two conditions are explicitly stated in polynomially bounded primitive recursion, they are hidden in bounded primitive recursion on binary notation. More precisely, the restriction to polynomially many steps is hidden in the fact that we are actually recurring not on y but on its binary representation (hence on $|y|$). The restriction to polynomially bounded output lengths is instead hidden in the bound of the values (which are all, inductively, of the form $2^{p(|\vec{x}|,|y|)}$, and hence of polynomial length).

Rate of growth

The polynomial time computable functions are defined in terms of the time needed to compute them. We now take a different point of view, and look at their possible rates of growth.

The idea is simple: since the values of a polynomial time computable function are bounded in length by a polynomial, they must be bounded in size by a polyexponential. The next result just notices that any polyexponential can be bounded by a finite iteration of the weak exponential $x^{|y|}$ on x (iterations on y would not produce anything new, since e.g. $2^{|2^{|x|}|} = 2^{|x|+1}$).

Theorem VIII.2.16 A Skeleton for PF. *Let*

$$\mathrm{wexp}_0(x) = 2 \qquad \textit{and} \qquad \mathrm{wexp}_{n+1}(x) = (\mathrm{wexp}_n)^{|x|},$$

so that $\mathrm{wexp}_1 = 2^{|x|}$ *and* $\mathrm{wexp}_n(x) = (\cdots((2^{|x|})^{|x|})^{\cdots})^{|x|} = 2^{|x|^n}$. *Then:*

1. wexp_n *is polynomial time computable, for each* n
2. *every polynomial time computable function is dominated by some* wexp_n*, in the sense that given* f *polynomial time computable there is an* n *such that* $f(\vec{x}) \leq \mathrm{wexp}_n(\sum \vec{x})$*, for almost every* $\vec{x}$
3. *the diagonal function* $d_{\mathrm{PF}}(x) = \mathrm{wexp}_x(x)$ *dominates every polynomial time computable function, and it is thus not polynomial time computable.*

Proof. Part 1 is obvious, since the weak exponential function is polynomial time computable, and PF is closed under composition.

Part 3 follows from Part 2 and the monotonicity property:

- if $n \leq m$, then $\mathrm{wexp}_n(x) \leq \mathrm{wexp}_m(x)$,

since then $\mathrm{wexp}_n(x) \leq \mathrm{wexp}_x(x)$ for every n and $x \geq n$.

Part 2 follows from the observation that the values of a polynomial time computable function must be polynomially bounded in length, because in a polynomial number of moves a Turing machine can only write down a polynomial number of digits. Then the values themselves must be exponentially bounded in size, and hence bounded by some wexp_n.

More precisely, if f is polynomial time computable, there is a polynomial p such that $f(\vec{x})$ is computable in time $p(|\sum \vec{x}|)$, so that $|f(\vec{x})| \leq p(|\sum \vec{x}|)$ and

$$f(\vec{x}) \leq 2^{p(|\sum \vec{x}|)} \leq 2^{|\sum \vec{x}|^n} = \mathrm{wexp}_n(\sum \vec{x}),$$

for some n and almost every $\vec{x}$. □

Corollary VIII.2.17 *$2^{2^{|x|}}$ and 2^x are not polynomial time computable.*

Proof. Since $2^{|x|}$ dominates every $|x|^n$, $2^{2^{|x|}}$ dominates every wexp_n, and hence every polynomial time computable function. In particular, $2^{2^{|x|}}$ is not polynomial time computable.

Since $2^{2^{|x|-1}} \leq 2^x < 2^{2^{|x|}}$, it also follows that 2^x is not polynomially time computable. □

In particular, PF is closed under weak exponential, but not under exponential. To obtain a class of functions closed under exponential we have to go all the way up to elementary functions (see Section 7).

Feasibly computable functions ⋆

In Chapter I we have introduced a number of formal approaches to the informal notion of *effective computability*, which resulted in equivalent definitions of the class of recursive functions. This fact, together with the stability and absoluteness of the class, led to a formulation of the following thesis:

> **(Church [1936], Turing [1936])** *Every effectively computable function is recursive.*

The claim established the significance of the notion of recursiveness even though, as discussed in Section I.8, a closer investigation exposes first a variety of possible interpretations, and then the weakness (or lack) of evidence in favor of most of them.

There is, however, an aspect of the computations that has been disregarded completely in Chapter I, namely their *practical feasibility*. The stability of the class of polynomial time computable functions has led to a formulation of the following thesis:

> **(Cobham [1964], Edmonds [1965])** *Every feasibly computable function is polynomial time computable (in the length of the input).*

This claim has certainly helped to establish the significance of the notion of polynomial time computability. It is now our task to take a closer look at it.

a) Polynomial and nonpolynomial bounds

A few trivial calculations are enough to expose the fiction of taking polynomial bounds as significant upper bounds of feasible computations.

First, *the nonpolynomial function $|x|^{\log_{10}(\log_{10}|x|)}$ grows slower than the polynomial $|x|^2$ on any input that can be physically written down.* For example, since the number of particles of the universe has been estimated at around 10^{80}, it is safe to say that only inputs of smaller length can be written down. It is now enough to notice that $|x|^{\log_{10}(\log_{10}|x|)}$ catches up with $|x|^2$ only at $|x| = 10^{100}$.

Looking now at more practical lengths of input, the next table shows some revealing data.

$\|x\|$	$\|x\|^2$	$\|x\|^5$	$\|x\|^{10}$	$\|x\|^{\log_{10}\|x\|}$	$10^{\sqrt{\|x\|}}$	$2^{\|x\|}$	$10^{\|x\|}$
10	10^2	10^5	10^{10}	10	$10^{3.16}$	10^3	10^{10}
100	10^4	10^{10}	10^{20}	10^4	10^{10}	10^{30}	10^{100}
$1,000$	10^6	10^{15}	10^{30}	10^9	$10^{31.6}$	10^{300}	$10^{1,000}$
$10,000$	10^8	10^{20}	10^{40}	10^{16}	10^{100}	$10^{3,000}$	$10^{10,000}$

Thus, even without considering the slowly growing function $|x|^{\log_{10}(\log_{10}|x|)}$, one sees that *the faster growing functions* $|x|^{\log_{10}|x|}$ *and* $10^{\sqrt{|x|}}$ *grow slower than small degree polynomials on quite large inputs*. For example, $|x|^{\log_{10}|x|}$ catches up with $|x|^5$ at 10^5, and with $|x|^{10}$ at 10^{10}; and $10^{\sqrt{|x|}}$ catches up with $|x|^5$ at 10^2, and with $|x|^{10}$ at around 10^3. Now, by using lengths up to 10^2 one can already write down incredibly large numbers (up to 10^{100}). Beyond lengths such as 10^4, one certainly reaches limits of applicability, since a number of that length is of the order of a 1 followed by three pages of 0's. For example, the representation of two numbers of length 10^6 would fill up the present volume.

A better grasp of the significance of the discussion just given can be obtained by looking at the time needed for a computation, for example on a computer performing a million operations per second. Then an algorithm would take:

- *on inputs of length 100*
  0.01 second if working in time $|x|^2$ or $|x|^{\log_{10}|x|}$; 1 second if working in time $|x|^3$; 1.5 minutes if working in time $|x|^4$; 2 hours and 45 minutes if working in time $|x|^5$ or $10^{\sqrt{|x|}}$; 12 days if working in time $|x|^6$; and 3 years if working in time $|x|^7$.

- *on inputs of length 1,000*
  1 second if working in time $|x|^2$; 16.5 minutes if working in time $|x|^3$ or $|x|^{\log_{10}|x|}$; 12 days if working in time $|x|^4$; and 33 years if working in time $|x|^5$.

On the other hand, there is no doubt that *exponential functions behave wildly for small inputs*. For example, at 100, $10^{|x|}$ gives a bound that $|x|^{10}$ reaches only at 10^{10}.

Thus *exponential functions are certainly bad, but slowly growing nonpolynomial functions are better, even for large inputs, than polynomials with even moderate exponents*.

b) Significance of polynomial upper bounds

Since, by definition, an algorithm runs in polynomial time if there is a polynomial $|x|^n$ that bounds its time complexity for almost every input, finitely many arguments are disregarded as insignificant. This is certainly sensible when the stress is on infinitistic behavior, as in a mathematical theory of functions. But it becomes ludicrous when the concern shifts to practical matters, since by allowing finitely many exceptions one may classify an algorithm as running in polynomial time because its behavior is good on inputs that one would never consider in practice, without saying anything about the significant inputs.

c) Significance of nonpolynomial lower bounds

Since, by definition, an algorithm does not run in polynomial time if for any n, there are infinitely many inputs x such that the algorithm takes on x more time than $|x|^n$, infinitely many worst-case inputs are regarded as significant. Again, this is certainly sensible when the stress is on a global behavior, but it becomes ludicrous when the concern shifts to practical matters, since one may classify an algorithm as not running in polynomial time because its behavior is bad on infinitely many inputs that one would never consider in practice.

This may occur in two different ways: either because of *size* (the smallest worst-case inputs being greater than all the inputs of practical significance), or because of *distribution* (the worst-case inputs falling in different regions than the inputs of practical significance)

d) An example: Linear Programming

Dantzig [1949] provided an algorithm for the solution of Linear Programming, called the *simplex method*. This has been extensively used in practice ever since, with impressive *empirical* evidence of running fast on inputs of interest.

Klee and Minty [1972] and Zadeh [1973] have proved that, according to the complexity-theoretical classification, the simplex algorithm is exponential time computable, and thus badly nonpolynomial.

Khachian [1979] has found another algorithm for the solution of Linear Programming, called the *ellipsoid method*, that is polynomial time computable. Despite the fact that the algorithm actually runs in quadratic time, the coefficients of the quadratic bound are large, and in practice the algorithm behaves worse than the simplex method.

Smale [1983] has finally proved that, on a set of measure 1 (for the appropriate probability space), the simplex method is actually linear, with small coefficients. This provides a *theoretical* explanation of the practical efficiency of the simplex method, despite its provable inefficiency in terms of complexity-theoretical classifications.

For history and a treatment of this topics see Dantzig [1963], and Papadimitriou and Steiglitz [1982].

e) Conclusion

The discussion above shows that the crude classification of feasible computability in terms of polynomial upper bounds on almost every input misses the mark. *Practical computability is concerned with the behavior of specific algorithms, running well* (but not necessarily in polynomial time on almost all inputs) *on specific machines* (certainly not Turing machines) *and on finitely many inputs of a specific structure* (not necessarily captured by a purely numerical bound on the length of their representation).

The concerns of feasible computability are thus orthogonal to those of Complexity Theory. The latter should be taken as a study that tells us a lot about the abstract world of the recursive functions, but little about real-life issues concerning computations.

Nevertheless, we can conclude as in Section I.8 about recursiveness: the notion of polynomial time computability has more than sufficient motivations (discussed in this section) to deserve a thorough mathematical study, disregarding its connections with feasibility, which are seen here as a distraction.

The class P

We turn now to the consideration of one of the best known complexity classes.

Definition VIII.2.18 **P** *is the class of sets computable in deterministic polynomial time in the length of the input.*

Complementing similar machine-theoretical characterizations of CONSPACE and LOGSPACE, Cook [1971a] and Chandra and Stockmeyer [1976] have proved that P is the class of sets computable by (non)deterministic two-way *multihead pushdown automata*, as well as the class of sets acceptable by two-way *multihead alternating automata*.

P is defined in terms of polynomially bounded *time* computations. Chandra, Kozen and Stockmeyer [1981] have characterized it in terms of logarithmically bounded *space* computations, by showing that P = ALOGSPACE, i.e. P is the class of sets computable in logarithmic space in the length of the input by alternating Turing machines.

If one defines the classes

$$P_n = \text{TIME}\,[\,|x|^n\,]$$

(in particular, P_1 is the class of sets computable in real time), which are not independent of the model of computation, then one immediately has the first of a long series of hierarchy theorems.

Theorem VIII.2.19 Hierarchy Theorem for P.

1. $\mathrm{P}_n \subset \mathrm{P}_{n+1}$, *for each* n
2. $\mathrm{P} = \bigcup_{n\in\omega} \mathrm{P}_n$.

Proof. Part 1 follows from the Time Hierarchy Theorem VII.4.15. Part 2 is just a restatement of the definition of P. $\Box$

Since P is a deterministic class, it is trivially closed under complements.

Theorem VIII.2.20 (Cook [1974]) *There are* log-*space complete sets for* P.

Proof. We would like to proceed as in VIII.1.18, and define

$$\langle x, e, n\rangle \in \mathcal{K}_0^{\mathrm{P}} \;\Leftrightarrow\; M_e \text{ accepts } x \text{ in time } |x|^n.$$

But if we try to show that $\mathcal{K}_0^{\mathrm{P}} \in \mathrm{P}$, then we run into the problem of having different polynomial bounds for different inputs, so that in general there is no fixed polynomial bound that works for all inputs.

The unexpected twist is that the set defined by

$$\langle x, e, n\rangle \in \mathcal{K}_0^{\mathrm{P}} \;\Leftrightarrow\; M_e \text{ accepts } x \text{ in time } |n|$$

or, similarly,

$$\langle x, e, 2^n\rangle \in \mathcal{K}_0^{\mathrm{P}} \;\Leftrightarrow\; M_e \text{ accepts } x \text{ in time } n$$

actually works.

- $\mathcal{K}_0^{\mathrm{P}} \in \mathrm{P}$
  Given an input $\langle x, e, n\rangle$ we decode it into x, e and n, and copy n on a tape to be used as a counter of length $|n|$. We then simulate M_e on input x for $|n|$ moves, using a working tape.

  The whole procedure requires only polynomial time, since the decoding procedure works in log-space, and is such that $|n| \leq |\langle x, e, n\rangle|$.

- *if* $B \in \mathrm{P}$, *then* $B \leq_m^{\mathrm{LOGSPACE}} \mathcal{K}_0^{\mathrm{P}}$
  Let M_e be a deterministic Turing machine computing B in time $|x|^n$ for

some n, and let a be the constant loss needed to simulate M_e on a fixed Turing machine M for $\mathcal{K}_0^{\mathrm{P}}$. Then

$$\begin{aligned} x \in B \quad &\Leftrightarrow \quad M_e \text{ accepts } x \text{ in time } |x|^n \\ &\Leftrightarrow \quad M \text{ accepts } x \text{ in time } a \cdot |x|^n \\ &\Leftrightarrow \quad \langle x, e, 2^{a \cdot |x|^n} \rangle \in \mathcal{K}_0^{\mathrm{P}}. \end{aligned}$$

It remains to show that the function $f(x) = \langle x, e, 2^{a \cdot |x|^n} \rangle$ is log-space computable. Then from

$$x \in B \;\Leftrightarrow\; f(x) \in \mathcal{K}_0^{\mathrm{P}}$$

we have $B \leq_m^{\mathrm{LOGSPACE}} \mathcal{K}_0^{\mathrm{P}}$.

Since the coding procedure works in log-space and x, e and n are given, the only point to argue about is that we can write down $2^{a \cdot |x|^n}$, i.e. a 1 followed by $a \cdot |x|^n$ 0's, in log-space. For this we only need a binary counter of length $\log a \cdot |x|^n = \log a + n \cdot \log |x|$. Thus f is computable in space $\log a + n \cdot \log |x|$ and hence, by the Linear Speed-Up Theorem for space VII.4.8, in space $\log |x|$. $\Box$

Notice that $\mathcal{K}_0^{\mathrm{P}}$ belongs to P_1 by definition. However, there is no contradiction with the Hierarchy Theorem VIII.2.19 for P, because P_1 is *not* closed under log-space reductions. Indeed, by the Space-Time Trade-Off Theorem VII.4.17 a function computable in space $\log |x|$ is only computable in time $c^{\log |x|}$ for some c, and hence in time $|x|^a$ for some a (namely, for $a = \log c$).

What the Hierarchy Theorem shows is that there can be no fixed polynomial bound of the computation time of reductions of sets in P to $\mathcal{K}_0^{\mathrm{P}}$ (or, more generally, that there can be no fixed bound of the complexities of both a complete set for P and the reductions of sets in P to it). What the computational simplicity of $\mathcal{K}_0^{\mathrm{P}}$ shows is that the complexity of a set $B \in \mathrm{P}$ (measured by the least n such that $B \in \mathrm{P}_n$) is mirrored in the best complexity of its reductions to $\mathcal{K}_0^{\mathrm{P}}$.

Examples of real-life problems that turn out to be log-space complete for P are given in Jones and Laaser [1976], Garey and Johnson [1979], Wagner and Wechsung [1986], Johnson [1990], Papadimitriou [1994], Greenlaw, Hoover and Ruzzo [1995]. Two significant examples are:

- *Horn Satisfiability*, i.e. truth-value satisfiability of Horn formulas of the Classical Propositional Calculus (Jones and Laaser [1976])
- *Linear Programming*, i.e. satisfiability over the rational (or real) numbers of systems of linear inequalities with integer coefficients (Khachian [1979], Dobkin, Lipton and Reiss [1979]).

Exercise VIII.2.21 *VIII.2.20 is best possible, in the sense that if* $\lim_{x\to\infty} \frac{s(x)}{\log|x|} = 0$ *then there is no set complete for* P *w.r.t. m-reductions using functions computable in space s.* (Hint: suppose A is computable in time $|x|^n$ for some n, f is computable in space s, and

$$x \in B \ \Leftrightarrow\ f(x) \in A.$$

By the proof of VII.4.5, $f(x)$ is computable in time

$$|x| \cdot c^{s(x)} = |x| \cdot (2^{\log c})^{s(x)} = |x| \cdot 2^{(\log c)\cdot s(x)},$$

for some constant c. By the hypothesis on s, $(\log c) \cdot s(x) \leq \log|x|$ for almost every x, and thus $f(x)$ is computable in time $|x|^2$. Then there is a fixed polynomial p, depending only on A, such that any such set B is computable in time p. But, by the Time Hierarchy Theorem VII.4.15, there are sets $B \in$ P not computable in time p.)

Although the previous exercise rules out the possibility of defining significant reducibilities stronger than $\leq_m^{\text{LOGSPACE}}$ in terms of space bounds, this does not mean than such stronger reducibilities cannot be defined in other ways. For examples, see Jones, Lien and Laaser [1976], and Cook and McKenzie [1987].

Logarithmic space again and beyond ⋆

As we have already noted above, from the general Space-Time Trade-Off Theorem VII.4.17 we immediately have that

$$\text{LOGSPACE} \subseteq \text{P},$$

because a set computable in space $\log|x|$ is also computable in time $c^{\log|x|}$ for some c, and hence in time $|x|^a$ for some a (namely, for $a = \log c$).

It is not known whether the inclusion LOGSPACE $\subseteq$ P is proper. Since P $=$ ALOGSPACE, this is equivalent to the problem of whether the inclusion

$$\text{LOGSPACE} \subseteq \text{ALOGSPACE}$$

is proper, i.e. whether it is possible to simulate alternating Turing machines by usual Turing machines without space loss. A positive answer would provide an analogue of VII.4.6.1, which shows that it is possible to simulate multitape Turing machines by one-tape Turing machines without space loss.

As for NLOGSPACE, one can first use Savitch's Theorem VII.4.20 to claim that a set acceptable in nondeterministic space $\log|x|$ is also computable in deterministic space $(\log|x|)^2$, but then the Space-Time Trade-Off Theorem only states that such a set is computable in time $c^{(\log|x|)^2}$ for some c, and hence in time $|x|^{a\cdot\log|x|}$ for some a. Since this falls short of being a polynomial bound, the next result is thus not trivial.

Proposition VIII.2.22 (Cook [1971a]) NLOGSPACE $\subseteq$ P.

Proof. By VII.4.25.2. □

Once again, it is not known whether the inclusion is proper. However, by slightly extending both classes LOGSPACE and NLOGSPACE by a consideration of the less restricted space bounds suggested by the failed proof of VIII.2.22 attempted above, we do reach a class that is known to be different from P, as we now see.

Definition VIII.2.23 POLYLOGSPACE *is the class of sets computable in deterministic space* $\log^n$ *in the length of the input, for some* n*.*

POLYLOGSPACE is defined in terms of bounded *space* computations. Chandra, Kozen and Stockmeyer [1981] have characterized it in terms of bounded *time* computations, by showing that POLYLOGSPACE = APOLYLOG, i.e. POLYLOGSPACE is the class of sets computable by alternating Turing machines in time $\log^n$ in the length of the input, for some n.

If one defines the classes

$$\text{LOGSPACE}^n = \text{SPACE}\,[\,(\log |x|)^n\,]$$

(in particular, LOGSPACE^1 = LOGSPACE), and

$$\text{NLOGSPACE}^n = \text{NSPACE}\,[\,(\log |x|)^n\,]$$

(in particular, NLOGSPACE^1 = NLOGSPACE), then one immediately has a double hierarchy theorem.

Theorem VIII.2.24 Hierarchy Theorem for POLYLOGSPACE.

1. LOGSPACE$^n \subset$ LOGSPACE^{n+1}, *for each* n
2. NLOGSPACE$^n \subset$ NLOGSPACE^{n+1}, *for each* n
3. POLYLOGSPACE $= \bigcup_{n\in\omega}$ LOGSPACE$^n = \bigcup_{n\in\omega}$ NLOGSPACE^n.

Proof. Parts 1 and 2 follow from the Space Hierarchy Theorems VII.4.13 and VII.4.24, respectively.

The fist equality of Part 3 is simply a restatement of the definition of POLYLOGSPACE. The second equality follows from the first and from Savitch's Theorem VII.4.20, because

$$\text{LOGSPACE}^n \subseteq \text{NLOGSPACE}^n \subseteq \text{LOGSPACE}^{2n}.$$

In other words, the unions of the deterministic and nondeterministic classes coincide. □

It is not known whether LOGSPACE^n = NLOGSPACE^n for some n (for $n = 1$, this reduces to the problem of whether LOGSPACE = NLOGSPACE). Notice that this might fail for every $n \geq 1$, despite the fact that

$$\bigcup_{n\in\omega} \text{LOGSPACE}^n = \bigcup_{n\in\omega} \text{NLOGSPACE}^n.$$

As usual for deterministic complexity classes, POLYLOGSPACE is trivially closed under complements. The next result is instead unusual for complexity classes.

Proposition VIII.2.25 (Book [1976]) *There is no* log*-space complete set for* POLYLOGSPACE*.*

Proof. Given a log-space complete set A for POLYLOGSPACE, there would be n such that $A \in$ LOGSPACE^n. We reach a contradiction by proving

$$\begin{aligned} \text{POLYLOGSPACE} &\subseteq \text{LOGSPACE}^n \\ &\subset \text{LOGSPACE}^{n+1} \\ &\subseteq \text{POLYLOGSPACE}. \end{aligned}$$

The strict inclusion follows from the Hierarchy Theorem VIII.2.24, and the last inclusion holds by definition. Finally, the first inclusion holds because every set in POLYLOGSPACE is log-space m-reducible to A by completeness of A, and LOGSPACE^n is downward closed under log-space m-reducibility (see VIII.1.16.a). □

One should compare the previous proof to the discussion after VIII.2.20. The main reason why the argument works here but a similar argument would not work there, is that LOGSPACE^n is closed under log-space reductions but P$_n$ is not.

Corollary VIII.2.26 P $\neq$ POLYLOGSPACE.

Proof. By VIII.2.20 and VIII.2.25. □

Notice that the proof of the corollary is indirect, and it does not establish whether any of the two classes is contained in the other. The sets complete for P provide natural candidates of sets not in POLYLOGSPACE.

It is not known whether LOGSPACE$^2 \subseteq$ P, and this is the reason why the proof of VIII.2.22 could not simply make use of Savitch's Theorem VII.4.20. The sets complete for LOGSPACE^n ($n \geq 2$) provide natural candidates of sets not in P.

A look inside P $\star$

Since it is not known whether any of the two classes P and POLYLOGSPACE is contained in the other, it is natural to consider problems in their intersection.

Steven Cook [1979] has defined SC (*Steve's Class*) as the class of sets computable by deterministic Turing machines working in polynomial time *and* polylogarithmic space. A natural hierarchy for SC is obtained by considering the classes SC_n of sets computable by deterministic Turing machines working in polynomial time and $\log^n$ space (in particular, SC_1 is the class of sets computable by Turing machines in logarithmic space).

Nicholas Pippinger [1979] has similarly defined NC (*Nick's Class*) as the class of sets acceptable by alternating Turing machines working in polylogarithmic time and logarithmic space (some care is needed in the definition, because of the small time bounds). A natural hierarchy for NC is obtained by considering the classes NC_n of sets acceptable by alternating Turing machines working in $\log^n$ time and logarithmic space (in particular, NC_1 is the class of sets computable by alternating Turing machines in logarithmic time).

Both SC and NC are quite independent of the model of computation, and admit a number of alternative characterizations. For example, NC can be defined either as the class of sets computable by parallel random access machines working in polylogarithmic time and with a polynomial number of processors, or as the class of sets computable by uniform families of Boolean circuits of polylogarithmic *depth* and polynomial size. Similarly, SC can be defined as the class of sets computable by uniform families of Boolean circuits of polylogarithmic *width* and polynomial size.

Such characterizations show that NC is a natural formalization of the notion of a set computable by a *feasible parallel algorithm*, exactly as P is a natural formalization of the notion of a set computable by a *feasible sequential algorithm*. Thus the problem of whether the inclusion $\mathrm{NC} \subseteq \mathrm{P}$ is proper is equivalent to the problem of whether there are feasible algorithms that are inherently sequential. A negative answer would show that any feasibly sequential algorithm can be massively parallelized.

Obviously, both SC and NC are contained in $\mathrm{P} \cap \mathrm{POLYLOGSPACE}$, but it is not known whether any of SC, NC and $\mathrm{LOGSPACE}^2$ is contained in the other, nor whether the hierarchies for SC and NC are proper. However,

$$\mathrm{NC}_1 \subseteq \mathrm{SC}_1 = \mathrm{LOGSPACE} \subseteq \mathrm{NLOGSPACE} \subseteq \mathrm{NC}_2 \subseteq \mathrm{LOGSPACE}^2.$$

In particular, $\mathrm{NLOGSPACE} \subseteq \mathrm{NC}$. It is not known whether $\mathrm{NLOGSPACE} \subseteq \mathrm{SC}$, nor whether any of the previous inclusions is proper. NC_2 is considered to be a formalization of the notion of a set computable by an *efficient parallel algorithm*.

For a survey of complexity classes inside P, see Johnson [1990] and Karp and Ramachandran [1990]. For a treatment, see Balcázar, Díaz and Gabarró [1990], and Papadimitriou [1994].

Polynomial time degrees $\star$

The notion of log-space reducibility introduced in VIII.1.15 is useful in the treatment of complete sets, especially for small complexity classes such as NLOGSPACE and P (see VIII.1.18 and VIII.2.20). Satisfactory analogues of the reducibilities introduced in Chapter III must however be based on polynomial time computability, which is the correct analogue of the notion of recursiveness on which such reducibilities are based.

The simplest case of polynomial time reducibility is the following analogue of m-reducibility.

Definition VIII.2.27 (Karp [1972], Levin [1973a]) *A is* **polynomial time m-reducible** *to B ($A \leq_m^{\mathrm{P}} B$) if, for some function f computable in polynomial time in the length of the input,*

$$x \in A \Leftrightarrow f(x) \in B.$$

A is **polynomial time m-equivalent** *to B ($A \equiv_m^{\mathrm{P}} B$) if $A \leq_m^{\mathrm{P}} B$ and $B \leq_m^{\mathrm{P}} A$.*

Note that $\leq_m^{\mathrm{P}}$ is a reflexive and transitive relation, and thus $\equiv_m^{\mathrm{P}}$ is an equivalence relation.

Definition VIII.2.28 *The equivalence classes of sets w.r.t. polynomial time m-equivalence are called* **polynomial time m-degrees***, and* $(\boldsymbol{\mathcal{D}_m^{\mathrm{P}}}, \leq)$ *is the structure of polynomial time m-degrees, with the partial ordering $\leq$ induced on them by $\leq_m^{\mathrm{P}}$.*

$\mathbf{0}_m^{\mathrm{P}}$ *is the least polynomial time m-degree, containing exactly the polynomial time computable sets different from $\emptyset$ and ω.*

Polynomial time analogues of the other strong reducibilities defined in Chapter III, in particular

$$\leq_1^{\mathrm{P}} \quad \leq_c^{\mathrm{P}} \quad \leq_d^{\mathrm{P}} \quad \leq_p^{\mathrm{P}} \quad \leq_{tt}^{\mathrm{P}} \qquad \text{and} \qquad \leq_{bc}^{\mathrm{P}} \quad \leq_{bd}^{\mathrm{P}} \quad \leq_{bp}^{\mathrm{P}} \quad \leq_{btt}^{\mathrm{P}},$$

are defined in Ladner, Lynch and Selman [1975] and Selman [1982]. See also VIII.3.10 and VIII.3.11 for a polynomial time analogue $\equiv^{\mathrm{P}}$ of recursive isomorphism types, and Selman [1978] for a polynomial time analogue $\leq_e^{\mathrm{P}}$ of the enumeration reducibility introduced on p. I.197 and studied in Chapter XIV.

The most fundamental case of a polynomial time reducibility is the analogue of Turing reducibility. It could be introduced by relativizing the class of polynomial time computable functions as in II.3.1 (see Melhorn [1976]), or by using a polynomial time analogue of the partial recursive functionals as in II.3.7 (see Kapron and Cook [1996]). We use instead a polynomial time analogue of the *oracle Turing machine*, mentioned in passing on p. I.197, and consisting of:

- an additional *query tape*, on which numbers can be written during a computation
- a special *query state* that, when entered, provides a yes or no answer, according to whether the number currently scanned on the query tape is or is not in the oracle
- additional *query instructions* of the form $q_a i q_b q_c$, telling the machine in query state q_a to enter state q_b or q_c, according to the answer of the oracle, i.e. according to whether $i = 0$ or $i = 1$.

The notion of acceptable system of indices for oracle Turing machines can be defined as in II.5.2, and in the following we let $\{M_e^X\}_{e\in\omega}$ and $\{N_e^X\}_{e\in\omega}$ be any such acceptable system for deterministic and nondeterministic oracle Turing machines. We also let $\{e\}^X$ be the partial function recursive in X computed by M_e^X.

Time and space bounds on oracle computations are defined as for usual multitape Turing machines (see p. 71), by considering the query tape as an additional working tape. Thus a space or time bound on a computation imposes a bound on the length of numbers that can be queried during it.[2]

A call to the oracle counts as a single move, independently of the query, because a single query instruction is needed to answer it. This accords with the intuition that the oracle is a device external to the computation itself, and gives an answer without any calculation.

Definition VIII.2.29 (Cook [1971]) *A is* **polynomial time Turing reducible** *to B ($A \leq_T^{\mathrm{P}} B$) if, for some index e and polynomial p:*

- $c_A \simeq \{e\}^B$
- *for all oracle X and input x, $\{e\}^X(x)$ is computable in time $p(|x|)$.*

[2]Notice that this does not produce a satisfactory log-space version of an oracle computation, because it forbids to answer the question of whether x is in the oracle by directly quering the oracle (one needs at least space $|x|$ to write down x on the query tape). Various versions of *log-space Turing reducibility* are proposed and discussed in Ladner and Lynch [1976], Lynch [1978], Savitch [1983], Ruzzo, Simon and Tompa [1984], Wilson [1985], Kirsig and Lange [1987], Buss [1987], [1988], Hartmanis, Chang, Kadin and Mitchell [1988].

Note that $\leq_T^{\mathrm{P}}$ is a reflexive and transitive relation, and thus $\equiv_T^{\mathrm{P}}$ is an equivalence relation.

Definition VIII.2.30 *The equivalence classes of sets w.r.t. polynomial time T-equivalence are called* **polynomial time $\boldsymbol{T}$-degrees***, and* $(\boldsymbol{\mathcal{D}_T^{\mathrm{P}}}, \leq)$ *is the structure of polynomial time T-degrees, with the partial ordering $\leq$ induced on them by $\leq_T^{\mathrm{P}}$.*

$\mathbf{0}_T^{\mathrm{P}}$ *is the least polynomial time T-degree, containing exactly the polynomial time computable sets.*

Exercises VIII.2.31 a) *If A is in* P*, then $A \leq_m^{\mathrm{P}} B$ for any $B \neq \emptyset, \omega$.*
b) *If $A \leq_T^{\mathrm{P}} B$ and B is in* P*, then so is A.*
c) *If $A \leq_m^{\mathrm{P}} B$, then $A \leq_T^{\mathrm{P}} B$.*

We now turn to a brief study of the structures $\boldsymbol{\mathcal{D}_m^{\mathrm{P}}}$ and $\boldsymbol{\mathcal{D}_T^{\mathrm{P}}}$, along the lines of Chapters V and VI and with the same conventions and notation. In particular, when dealing with polynomial time m-degrees we disregard the sets $\emptyset$ and ω.

Proposition VIII.2.32 *As partially ordered structures, $\boldsymbol{\mathcal{D}_m^{\mathrm{P}}}$ and $\boldsymbol{\mathcal{D}_T^{\mathrm{P}}}$ are uppersemilattices of cardinality $2^{\aleph_0}$ with a least but no maximal element. Moreover, each element has $2^{\aleph_0}$ successors and at most countably many predecessors.*

Proof. As in V.1.12 and VI.1.1. $\square$

The next result imposes a strong limitation on possible minimal polynomial time degrees.

Proposition VIII.2.33 (Homer [1987]) *If A has a minimal polynomial time m-degree or T-degree, then A is recursive.*

Proof. We prove that if A is nonrecursive, then the set B defined by

$$2^x \in B \Leftrightarrow x \in A$$

is such that $B \leq_m^{\mathrm{P}} A$ and $A \not\leq_T^{\mathrm{P}} B$. Since A is nonrecursive, B is nonrecursive and hence not in P. Thus B has a nonzero degree strictly below the degree of A.

To prove that $B \leq_m^{\mathrm{P}} A$, consider the function f defined by

$$f(z) = \begin{cases} x & \text{if } z = 2^x \\ a & \text{otherwise,} \end{cases}$$

where $a \notin A$ (a exists because A is certainly not equal to ω, since it is not recursive). Then

$$z \in B \Leftrightarrow f(z) \in A$$

by definition. The function f is polynomial time computable, because to compute $f(z)$ it is enough to see if z consists of a 1 followed by a sequence of 0's, and if so to compute the number of 0's in the sequence. Thus f provides a polynomial time m-reduction of B to A.

To prove that $A \not\leq_T^{\mathrm{P}} B$, suppose that $A \simeq \{e\}^B$ with polynomial time bound $p_e(|x|)$. We show by induction that A is recursive. Given x, we determine whether $x \in A$ by recursively computing $\{e\}^B(x)$. Notice that the oracle B can be queried only for inputs of length $\leq p_e(|x|)$. Since p_e is a polynomial, for almost every x

$$p_e(|x|) < 2^{|x|-1} \leq x,$$

i.e. the oracle B can be queried only for inputs of length smaller than x, and hence of value smaller than 2^x. Membership in B of such elements is determined by membership in A of elements smaller than x, which is known by induction hypothesis. We can thus compute $\{e\}^B(x)$ without any use of the oracle. $\square$

By the previous result, only recursive sets can have minimal polynomial time degrees. The next result rules out this case, too.

Theorem VIII.2.34 (Ladner [1975], Machtey [1975]) *There is no minimal polynomial time m-degree or T-degree.*

Proof. Given a recursive set $A \notin \mathrm{P}$, we build a set $B \notin \mathrm{P}$ such that $B \leq_m^{\mathrm{P}} A$ and $A \not\leq_T^{\mathrm{P}} B$ by satisfying the following requirements:

$$\begin{array}{lll} R_{2e} & : & B \not\simeq \{e\} \ \text{ with time bound } p_e \\ R_{2e+1} & : & A \not\simeq \{e\}^B \text{ with time bound } p_e. \end{array}$$

The idea to satisfy the requirements is very simple:

- To satisfy R_{2e} we just have to make B look like A long enough, from a certain point on. Since $A \notin \mathrm{P}$, we eventually hit a witness z such that $B(z) \not\simeq \{e\}(z)$ in time $p_e(|z|)$.

- To satisfy R_{2e+1} we just have to make B look empty long enough, from a certain point on. Since $A \notin \mathrm{P}$, we eventually hit a witness z such that $A(z) \not\simeq \{e\}^B(z)$ in time $p_e(|z|)$.

To ensure $B \leq_m^{\mathrm{P}} A$, we want a polynomial time computable relation R that tells us whether we are forcing B to look like A or $\emptyset$. In other words,

$$x \in B \ \Leftrightarrow \ x \in A \ \wedge \ R(x).$$

Then $B \leq_m^{\mathrm{P}} A$ via the following polynomial time computable function f:

$$f(x) = \begin{cases} x & \text{if } R(x) \\ a & \text{otherwise,} \end{cases}$$

where $a \notin A$ (a exists because A is certainly not equal to ω, since it is not in P). The idea to make R polynomial time computable is to change its values only at arguments of the form 2^x, and to let it be constant among such arguments. If we make $R(2^x)$ polynomial time computable in x, then $R(z)$ turns out to be polynomial time computable in $|z|$.

The construction is the following, where $g(s)$ is the requirement that we are trying to satisfy at stage s. We start by letting $R(0)$ and $R(1)$ be false and $g(0) = 0$. At stage $s > 0$, we define $R(2^s)$ and $g(s)$ by the following procedure. We simulate s moves of the previous construction, and let n be the greatest number such that $R(2^n)$ and $g(n)$ can be determined by this simulation.

- If $g(n) = 2e$, then at stage n we were trying to satisfy R_{2e}. To see if we succeeded, we make s moves of a dovetailed computation of

  $$B(0), \{e\}(0), B(1), \{e\}(1), \ldots$$

  This requires simulating the whole construction of B, in particular using the oracle A in the intervals in which B looks like it.

  If we find an element z such that $B(z) \not\simeq \{e\}(z)$, then R_{2e} has been satisfied and we can turn to the next requirement, by letting $R(2^s)$ be false (i.e. B starts looking empty) and $g(s) = 2e + 1$.

  Otherwise, we do not know whether R_{2e} has been satisfied and we continue to pursue our strategy for it, by letting $R(2^s)$ be true (i.e. B continues to look like A) and $g(s) = 2e$.

- It $g(n) = 2e + 1$, then at stage n we were trying to satisfy R_{2e+1}. To see if we succeeded, we make s moves of a dovetailed computation of

  $$A(0), \{e\}^B(0), A(1), \{e\}^B(1), \ldots$$

  This again requires simulating the whole construction of B, using the oracle A in the intervals in which B looks like it, as well as to compute the needed values of A.

  If we find an element z such that $A(z) \not\simeq \{e\}^A(z)$, then R_{2e+1} has been satisfied and we can turn to the next requirement, by letting $R(2^s)$ be true (i.e. B starts looking like A) and $g(s) = 2e + 2$. Otherwise, we do not know whether R_{2e+1} has been satisfied and we continue to pursue our strategy for it, by letting $R(2^s)$ be false (i.e. B continues to look empty) and $g(s) = 2e + 1$.

Notice that when a requirement is attacked g takes as value the index of the requirement, while it increases by 1 when the requirement is satisfied. To see that all requirements are satisfied, it is thus enough to show that g is onto. Suppose i is the least number such that $i+1$ is not in the range of g. Then, from a certain point on, the construction keeps on trying to satisfy R_i and it eventually satisfies it, which is a contradiction. $\Box$

The method of proof used in the previous result, called the *looking-back technique*, is very useful in situations where we cannot define functions by relying on a complete knowledge of its values for smaller arguments. In this case we simulate the previous construction for a permissible number of steps, thus gaining a partial knowledge of previous values and acting on it. The crucial point is that any piece of knowledge can be recovered, at sufficiently large stages.

Landweber, Lipton and Robertson [1981] and Chew and Machtey [1981] have introduced a *structural approach*, which is a sort of reversed looking-back technique. In this case one precomputes a function that, when iterated, gives intervals long enough to permit diagonalization (the looking-back technique finds these intervals along the way). See Schöning [1982], Regan [1983], [1992] and Ambos-Spies [1986] for codified versions of this structural approach.

Corollary VIII.2.35 *$\boldsymbol{\mathcal{D}}_{\boldsymbol{m}}^{\mathrm{P}}$ and $\boldsymbol{\mathcal{D}}_{\boldsymbol{T}}^{\mathrm{P}}$ are not elementarily equivalent to any of $\boldsymbol{\mathcal{D}}_{\boldsymbol{1}}$, $\boldsymbol{\mathcal{D}}_{\boldsymbol{m}}$, $\boldsymbol{\mathcal{D}}_{\boldsymbol{tt}}$, $\boldsymbol{\mathcal{D}}_{\boldsymbol{wtt}}$ and $\boldsymbol{\mathcal{D}}$.*

Proof. $\boldsymbol{\mathcal{D}}_{\boldsymbol{m}}^{\mathrm{P}}$ and $\boldsymbol{\mathcal{D}}_{\boldsymbol{T}}^{\mathrm{P}}$ differ from all the quoted structures because they do not have minimal degrees, while all the latter do, by VI.5.2, VI.2.2, VI.5.5 and V.5.11. $\Box$

Although we only stated the corollary for the most important degrees structures introduced so far, the lack of minimal degrees provides an elementary difference with all the degree structures we consider in this book, with only one exception: the enumeration degrees, for which the same result is proved by a similar two-step proof (see XIV.2.7 and XIV.2.9). However, elementary differences with $\boldsymbol{\mathcal{D}}_{\boldsymbol{e}}$ are provided by distributivity (VIII.2.36 and XIV.3.14) and density (VIII.2.39.b and Cooper [1990a]), discussed below.

The results proved so far apply to both structures $\boldsymbol{\mathcal{D}}_{\boldsymbol{m}}^{\mathrm{P}}$ and $\boldsymbol{\mathcal{D}}_{\boldsymbol{T}}^{\mathrm{P}}$. The next result provides instead an elementary difference between them.

Proposition VIII.2.36 (Ambos-Spies [1984c]) *$\boldsymbol{\mathcal{D}}_{\boldsymbol{m}}^{\mathrm{P}}$ is distributive, but $\boldsymbol{\mathcal{D}}_{\boldsymbol{T}}^{\mathrm{P}}$ is not.*

Proof. The proof that $\boldsymbol{\mathcal{D}}_{\boldsymbol{m}}^{\mathrm{P}}$ is distributive is the same as in VI.1.8.

To prove that $\boldsymbol{\mathcal{D}}_{\boldsymbol{T}}^{\mathrm{P}}$ is not distributive, we refer to Definition VI.1.3 and build sets A, B and C such that

$$A \leq_T^{\mathrm{P}} B \oplus C$$

and, for every B_0 and C_0,

$$A \equiv_T^{\mathrm{P}} B_0 \oplus C_0 \;\Rightarrow\; B_0 \not\leq_T^{\mathrm{P}} B \;\vee\; C_0 \not\leq_T^{\mathrm{P}} C.$$

The first condition is automatically ensured by letting

$$x \in A \;\Leftrightarrow\; x \in B \;\vee\; x \in C.$$

For the second condition, it is enough to restrict attention to sets $B_0, C_0 \leq_T^{\mathrm{P}} A$. We thus satisfy the following requirements, for every e, m, n, i, j:

$$\begin{array}{rcl} R_{e,m,n,i,j} & : & A \not\simeq \{e\}^{A_i \oplus A_j} \;\vee\; A_i \not\simeq \{m\}^B \;\vee\; A_j \not\simeq \{n\}^C \\ & & \text{with time bounds } p_e,\ p_m \text{ and } p_n, \end{array}$$

where $A_i \simeq \{i\}^A$ and $A_j \simeq \{j\}^A$ with time bounds p_i and p_j play the roles of B_0 and C_0.

We build A, B and C by stages. At stage $s+1$, suppose A_s, B_s and C_s are defined for every number of length smaller than $l(s)$. Pick a number x of length $l(s)$ and use $A_{i,s} \simeq \{i\}^{A_s}$ and $A_{j,s} \simeq \{j\}^{A_s}$ as approximations of B_0 and C_0. First, we compute the following numbers:

- $u = p_e(l(s))$, which is a bound on the possible queries on A_i and A_j
- $l(s+1) = \max\{p_i(u), p_j(u), p_m(u), p_n(u)\}$, which is a bound on all possible queries on A, B and C.

Then, we consider the following three conditions:

1. $A_s(x) \simeq \{e\}^{A_{i,s} \oplus A_{j,s}}(x)$
2. $A_{i,s} \simeq \{m\}^{B_s}$ up to u
3. $A_{j,s} \simeq \{n\}^{C_s}$ up to u.

If at least one condition fails, then we can satisfy $R_{e,m,n,i,j}$ by preserving the current situation, i.e. by freezing everything up to $l(s+1)$. If all three conditions hold, then we can satisfy $R_{e,m,n,i,j}$ by letting $A_{s+1} = A_s \cup \{x\}$, i.e. by putting x into A, and by choosing to put x into B or C according to the following cases:

- If $A_{i,s+1} = A_{i,s}$ and $A_{j,s+1} = A_{j,s}$, by condition 1 we have ensured

  $$A_{s+1}(x) \not\simeq A_s(x) \simeq \{e\}^{A_{i,s} \oplus A_{j,s}}(x) \simeq \{e\}^{A_{i,s+1} \oplus A_{j,s+1}}(x).$$

  Then it does matter if we put x into B or C. For example, we let $x \in B$.

- If $A_{i,s+1} \neq A_{i,s}$, the left-hand-side of condition 2 has changed and it is enough to freeze the right-hand-side, i.e. B. We thus put x into C, which keeps $B_{s+1} = B_s$.
- If $A_{j,s+1} \neq A_{j,s}$, the left-hand-side of condition 3 has changed and it is enough to freeze the right-hand-side, i.e. C. We thus put x into B, which keeps $C_{s+1} = C_s$. $\Box$

Corollary VIII.2.37 *$\boldsymbol{\mathcal{D}}_{\boldsymbol{m}}^{\mathrm{P}}$ and $\boldsymbol{\mathcal{D}}_{\boldsymbol{T}}^{\mathrm{P}}$ are not elementarily equivalent.*

The first-order theory of polynomial time degrees has the same degree (and actually the same isomorphism type) as the theory of Second-Order Arithmetic. This has been proved by Shinoda and Slaman [1990] for the polynomial time T-degrees, and by Nies for the polynomial time m-degrees.

We now turn from global to local theory. The most natural substructure of the polynomial time degrees consists of the degrees of recursive sets, which we briefly discuss.

Theorem VIII.2.38 Density Theorem for Recursive Sets (Ladner [1975], Machtey [1975]) *The polynomial time m-degrees and T-degrees of recursive sets are dense.*

Proof. The proof is a simple modification of VIII.2.34. For example, given $C <_T^{\mathrm{P}} A$, we build a set B such that $B \leq_m^{\mathrm{P}} A$, $A \not\leq_T^{\mathrm{P}} B \oplus C$ and $B \oplus C \not\leq_T^{\mathrm{P}} C$.

To satisfy $A \not\leq_T^{\mathrm{P}} B \oplus C$, we make B look empty long enough, so that $B \oplus C$ looks like C long enough.

To satisfy $B \oplus C \not\leq_T^{\mathrm{P}} C$, we rely on the hypothesis that $C \leq_T^{\mathrm{P}} A$ and make B look like A long enough, so that $B \oplus C$ looks like A long enough.

The construction succeeds because, by hypothesis, $C <_T^{\mathrm{P}} A$. More precisely, $A \not\leq_T^{\mathrm{P}} C$ ensures that the diagonalizations eventually succeed, while $C \leq_T^{\mathrm{P}} A$ ensures that C can be simulated in polynomial time from A. $\Box$

Exercises VIII.2.39 a) **Splitting and Density Theorem for Recursive Sets** (Ladner [1975]) *Every nonzero polynomial time m-degree (T-degree) of recursive sets splits above any given lesser polynomial time m-degree (T-degree).* (Hint: if $C <_T^{\mathrm{P}} A$, we build two disjoint sets B_0 and B_1 such that $B_0 \cup B_1 = A$, $B_i \leq_m^{\mathrm{P}} A$, $B_i \not\leq_T^{\mathrm{P}} C$ and $A \not\leq_T^{\mathrm{P}} B_i \oplus C$, by making B_1 look like A when B_0 looks like $\emptyset$ and conversely.)

b) **Density Theorem for All Sets** (Shinoda) *The polynomial time T-degrees of all sets are dense.* (Hint: if A is not recursive, we can no longer claim that the relation R produced by the proof of VIII.2.38 is polynomial time computable, and hence that $B \leq_m^{\mathrm{P}} A$. However, R is still polynomial time computable in A because so is C, and hence $B \leq_T^{\mathrm{P}} A$.)

Proposition VIII.2.40 (Ambos-Spies [1984c]) *The polynomial time m-degrees and T-degrees of recursive sets are not elementarily equivalent.*

Proof. The sets produced in the proof of VIII.2.36 are recursive, i.e. the polynomial time T-degrees of recursive sets are not distributive. $\square$

The first-order theory of polynomial time degrees of recursive sets has the same degree (and actually the same isomorphim type) as the theory of First-Order Arithmetic, in particular it is undecidable and not axiomatizable, while *the two-quantifier theory is decidable*. This has been proved by Shinoda and Slaman [1990] and Shore and Slaman [1992] for the polynomial time T-degrees, and by Nies (see also Ambos-Spies and Nies [1992a]) and Ambos-Spies, Fejer, Lempp and Lerman [1996] for the polynomial time m-degrees.

A number of other results about the polynomial time m-degrees and T-degrees have been proved by Ladner [1975], Machtey [1975], [1976], Melhorn [1976], Chew and Machtey [1981], Landweber, Lipton and Robertson [1981], Schöning [1982], [1984], Schmidt [1984], [1985], Ambos-Spies [1986], [1987], [1987a], Downey [1992], Ambos-Spies, Homer and Soare [1994]. See Ambos-Spies [1999] for a survey.

The theory of polynomial time degrees can be extended to a number of other subrecursive notions of degree, induced by relativization of other complexity classes such as the elementary or the primitive recursive functions (see Sections 7 and 8). Historically, however, things happened backwards: first were introduced the *primitive recursive degrees* (Kleene [1958], Axt [1959]), then the *elementary degrees* (Machtey [1975], Melhorn [1976]), and finally the *polynomial time degrees*. Most results proved or quoted above extend to any such notion of subrecursive degree, and various uniform abstract treatment have been attempted (see Basu [1970], Melhorn [1976], Schmidt [1985], Mueller [1991], Merkle [1996]).

For example, the proof of VIII.2.33 relies only on the fact that the recursive function 2^x majorizes every polynomial, and it has a polynomial time computable graph (i.e. we can determine whether $z = 2^x$ in time polynomial in $|z|$). Thus, the same proof works for any notion of degree induced by a complexity class for which there exists a recursive function majorizing every time bound for functions in the given class, and having a graph in the given class. Such a function automatically exists for any r.e. class of recursive functions with the Ritchie-Cobham property (see VII.3.8), in particular for the elementary or the primitive recursive functions (see VIII.7.6 and VIII.8.8). Together with the straightforward extension of the proof of VIII.2.34, this shows that *there is no minimal elementary or primitive recursive degree*.

The limits of VIII.2.33 are reached when all recursive functions are considered as time bounds. This is the case of truth-table degrees, since it follows from III.3.2 that $A \leq_{tt} B$ *if and only if* $A \leq_T B$ *with a recursive time bound* (in particular, the polynomial time reducibilities are special cases of truth-table reducibility). Indeed, the existence of minimal truth-table degrees (see

VI.5.5) proves that VIII.2.33 does not hold in this context. However, while the recursive functions cannot be majorized by a recursive function, they can be majorized by a function recursive in $\mathcal{K}$, and the proof of VIII.2.33 can be adapted to show that $\mathbf{0}'_{tt}$ does not have a minimal cover. Shore and Slaman [1992] have conjectured that the structures $\boldsymbol{\mathcal{D}}_{\boldsymbol{T}}^{\mathrm{P}}$ of polynomial time T-degrees and $\boldsymbol{\mathcal{D}}(\geq \mathbf{0}'_{tt})$ of truth-table degrees above $\mathbf{0}'_{tt}$ are isomorphic. As a partial result in this direction, notice that both structures are dense (by VIII.2.39.b and VI.5.4).

Another aspect of the proof of VIII.2.33 is that it relies on a polynomial time computable function f whose inverse is not polynomially bounded, because $f^{-1}(x) = 2^x$. Since polynomial time computable functions with a polynomially bounded inverse play an important role in Complexity Theory (see VIII.3.5), one can develop a theory of *polynomial time honest m-degrees* in which only such functions are allowed as reductions. Similarly for *polynomial time honest T-degrees*, in which the oracle cannot be asked questions whose length differs from the length of the input by more than a polynomial.

The proof of VIII.2.34 uses honest reductions, and thus *there is no minimal polynomial time honest m-degree or T-degree of recursive sets*. However, Homer [1987] has proved that if every polynomial time computable function with a polynomially bounded inverse has a polynomial time computable inverse (i.e. if P = NP, see VIII.3.8), then *minimal polynomial time honest T-degrees exist*, by a proof similar to V.5.11. Under the same hypothesis, Downey, Gasarch, Homer and Moses [1989] and Ambos-Spies [1989] have proved that *the structure $\boldsymbol{\mathcal{D}}_{\boldsymbol{hm}}^{\mathrm{P}}$ of polynomial time honest m-degrees resembles the structure $\boldsymbol{\mathcal{D}}_{\boldsymbol{m}}$ of m-degrees*, in the sense that analogues of VI.2.5 and X.5.11 hold. It is not known whether the sufficient hypothesis above is also necessary, in particular whether the existence of minimal polynomial time honest degrees implies P = NP.

For more about polynomial time honest degrees, see Basu [1970], Meyer and Ritchie [1972], Machtey [1972], [1974], [1975], [1975a], Young [1983], Homer [1985], [1987], Homer and Long [1987], Yang [1987], Downey, Gasarch, Homer and Moses [1989], Ambos-Spies [1989], Ambos-Spies, Homer and Yang [1990], Ambos-Spies and Yang [1990], Kristiansen [199?].

VIII.3 Nondeterministic Polynomial Time $\star$

The reader should be warned that Sections 3–5 are hypothetical, since they introduce classes of sets that are not known to be different from the class P previously discussed. This accounts for their classification as starred sections. Unless the reader has a direct interest in these matters, (s)he would do better to turn to the more reliable classes of functions and sets dealt with in the rest of the chapter.

The general opinion of workers in the area is that all classes in this and the next two sections are distinct. The fact, however, remains that a settlement of the relationships among these classes has eluded all efforts, and has thus become a major mathematical open problem.

The class NP

We now introduce what is probably the most intriguing class considered in Complexity Theory.

Definition VIII.3.1 (Bennett [1962], Edmonds [1965a], Cook [1971]) **NP** *is the class of sets acceptable in nondeterministic polynomial time in the length of the input.*

If one defines the classes

$$\mathrm{NP}_n = \mathrm{NTIME}\,[\,|x|^n\,]$$

(in particular, NP_1 is the class of sets acceptable in nondeterministic real time), which are not independent of the model of computation, then one obtains a hierarchy theorem similar to VIII.2.19. In particular,

$$\mathrm{NP} = \bigcup_{n \in \omega} \mathrm{NP}_n$$

is just a restatement of the definition of NP, while the fact that the hierarchy is proper has been proved by Cook [1973].

Deterministic polynomial time again $\star$

Since nondeterministic machines include the deterministic ones, $\mathrm{P} \subseteq \mathrm{NP}$. But it is not known whether the inclusion is proper.

By the Hierarchy Theorem for P, it is obvious that

$$\textit{for every } n \geq 1, \mathrm{P}_n \neq \mathrm{NP}.$$

Less obvious is that if one can get rid of a small amount of nondeterminism, then one can get rid of all of it, as we now show.

Proposition VIII.3.2 (Book [1972]) *If* $\mathrm{NP}_1 \subseteq \mathrm{P}$*, then* $\mathrm{P} = \mathrm{NP}$.

Proof. By induction from

$$\mathrm{NP}_n \subseteq \mathrm{P} \;\Rightarrow\; \mathrm{NP}_{2n} \subseteq \mathrm{P}.$$

This is proved by noting that $|x|^{2n} = (|x|^2)^n$ and thus, by quadratic padding, one can stretch a set accepted in nondeterministic time $|x|^{2n}$ to a set accepted in nondeterministic time $|x|^n$ (see VIII.6.6 for a similar argument). $\Box$

It follows that

$$\textit{for every } n \geq 1,\ \mathrm{P} \neq \mathrm{NP}_n.$$

Otherwise, suppose $\mathrm{P} = \mathrm{NP}_n$ for some n. We reach a contradiction by proving

$$\mathrm{P} \subseteq \mathrm{NP}_n \subset \mathrm{NP}_{n+1} \subseteq \mathrm{NP}_{2n} \subseteq \mathrm{P}.$$

The strict inclusion comes from the hierarchy result of Cook [1973] quoted above; the last inclusion holds by the proof of VIII.3.2 since, by half of the hypothesis, $\mathrm{NP}_n \subseteq \mathrm{P}$; and the first inclusion is the other half of the hypothesis.

Book and Greibach [1970] have proved that $\mathrm{P}_1 \neq \mathrm{NP}_1$ for the one-tape model of Turing machine that we are using, and Paul, Pippinger, Szemerédi and Trotter [1983] have extended the result to the multitape model of Turing machine. It is not known whether $\mathrm{P}_n = \mathrm{NP}_n$ for some n. Notice that this might fail for every $n \geq 1$, even if $\mathrm{P} = \mathrm{NP}$.

NP sets as analogues of r.e. sets: successes

We have already noted that, in general, nondeterministic complexity classes are subrecursive analogues of the class of r.e. sets. The correspondence between the classes of recursive and polynomial time computable functions discussed in the previous section makes the comparison between r.e. and NP sets even more compelling. We thus set out to prove for the latter analogues of the basic results proved in Section II.1 for the former.

By VIII.2.9, existential quantification of polynomial time computable predicates produces r.e. relations. To obtain NP relations it is necessary to restrict attention to *polynomially bounded* existential quantification of polynomial time computable predicates.

Theorem VIII.3.3 Normal Form Theorem for NP (Cook [1971], Karp [1972]) *An n-ary relation Q is in* NP *if and only if there is an $n+1$-ary polynomial time computable relation R and a polynomial p such that*

$$Q(\vec{x}) \ \Leftrightarrow\ (\exists y)_{|y| \leq p(|\vec{x}|)} R(\vec{x}, y).$$

Proof. If Q is accepted by a nondeterministic Turing machine N in polynomial time p, then

$$Q(\vec{x}) \ \Leftrightarrow\ (\exists y)(y \text{ codes an accepting computation of } N \text{ on } \vec{x}).$$

By arithmetization, as in VIII.2.8, the predicate in parenthesis is polynomial time computable. The code y of an accepting computation of N has length $\leq p(|\vec{x}|)$, and consists of configurations of length $\leq p(|\vec{x}|)$. Thus $|y| \leq (p(|\vec{x}|))^2$ and y has polynomial length.

Conversely, if

$$Q(\vec{x}) \;\Leftrightarrow\; (\exists y)_{|y| \leq p(|\vec{x}|)} R(\vec{x}, y),$$

then one can build a nondeterministic Turing machine N that on input $\vec{x}$ guesses y of length $\leq p(|\vec{x}|)$, and accepts $\vec{x}$ if and only if $R(\vec{x}, y)$. N works in polynomial time because R is polynomial time computable, and is nondeterministic because of the guess. $\Box$

Recall (see p. I.135) that Matiyasevitch [1970] has proved that the r.e. sets are exactly those of the form

$$y \in D \;\Leftrightarrow\; \exists \vec{x}[p(\vec{x}, y) = q(\vec{x}, y)],$$

where p and q are polynomials in $\vec{x}$ and y with coefficients in the natural numbers. Adleman and Manders [1976] (see also Jones and Matiyasevitch [1984]) investigate analogues of this strong normal form for NP sets, with bounded quantifiers. By the previous result, if the existential quantifier is polynomially bounded then the set is in NP. For the converse, only exponential bounds are known to suffice.

Kent and Hodgson [1982] characterize NP by allowing polynomial bounds on universal quantifiers and exponential bounds on existential quantifiers, and by considering all possible prefixes (a restricted form of the Arithmetical Hierarchy). Hodgson and Kent [1983] show that one such universal quantifier is enough.

We turn now from NP relations to NP sets. VIII.2.10.a shows that every nonempty r.e. set is the range of a polynomial time computable function. Notice that a function f computable in polynomial time p has polynomially bounded values, in the sense that

$$f(x) = y \;\Rightarrow\; |y| \leq p(|x|),$$

because f has to write down the output. By the previous result this is the right form of bound, but with the wrong order of variables. Indeed, since we are trying to characterize the range of f, we need a bound not on the values, but on the arguments. We thus look for functions f for which a polynomial q exists such that

$$f(x) = y \;\Rightarrow\; |x| \leq q(|y|).$$

This is an honesty condition, in the sense of VII.2.14. Indeed, suppose $f(x)$ is computable in time $p(|x|)$ for some polynomial p, which can be supposed to be monotone. Then $p(|x|) \leq p(q(|f(x)|))$, i.e. $f(x)$ is polynomial time computable in $|f(x)|$.

It turns out that the following notion suffices.

Definition VIII.3.4 *A function f has a* **polynomially bounded inverse** *if there is a polynomial q such that*

$$y \in \textit{range } f \ \Rightarrow\ (\exists x)_{|x| \leq q(|y|)}(f(x) = y),$$

i.e. such that $q(|y|)$ bounds the length of a counterimage of y via f.

Proposition VIII.3.5 Characterization of NP (Selman [1978]) *The following are equivalent:*

1. *$A \in \mathrm{NP}$*
2. *$A = \emptyset$ or A is the range of a polynomial time computable function with a polynomially bounded inverse.*

Proof. Let A be nonempty, and let $a \in A$. If $A \in \mathrm{NP}$, then by VIII.3.3 there is a polynomial p and a polynomial time computable relation R such that

$$x \in A \ \Leftrightarrow\ (\exists y)_{|y| \leq p(|x|)} R(x, y).$$

Then A is the range of

$$f(\langle x, y \rangle) = \begin{cases} x & \text{if } |y| \leq p(|x|) \ \wedge\ R(x,y) \\ a & \text{otherwise.} \end{cases}$$

Moreover, f is polynomial time computable because the case definition uses only a polynomial time computable predicate R and a check of length (which is polynomial time computable). The function f has a polynomially bounded inverse because if $f(\langle x, y \rangle) = x$, then

$$|\langle x, y \rangle| = 2(|x| + |y| + 1) \leq 2 \cdot (|x| + p(|x|) + 1)$$

by definition of coding (see the proof of VIII.1.18).

Conversely, let f be a polynomial time computable function such that

$$y \in \text{range } f \ \Rightarrow\ (\exists x)_{|x| \leq q(|y|)}(f(x) = y)$$

for some polynomial q. Then y is in the range of f if and only if

$$(\exists x)_{|x| \leq q(|y|)}(f(x) = y),$$

and this expression is in NP by VIII.3.3, because the matrix is polynomial time computable since f is. $\Box$

We now turn to a similar characterization of P.

Definition VIII.3.6 *A function f has a* **polynomial time computable inverse** *if there is a polynomial time computable function g such that*

$$y \in \mathit{range}\ f \ \Rightarrow\ f(g(y)) = y,$$

i.e. such that g picks up one counterimage of y via f.

Proposition VIII.3.7 Characterization of P. *The following are equivalent:*

1. $A \in \mathrm{P}$
2. *$A = \emptyset$ or A is the range of a polynomial time computable function with a polynomial time computable inverse.*

Proof. Let A be nonempty, and let $a \in A$. If $A \in \mathrm{P}$, then A is the range of

$$f(x) = \begin{cases} x & \text{if } x \in A \\ a & \text{otherwise.} \end{cases}$$

Moreover, f is polynomial time computable because A is, and the following is a polynomial time computable inverse for it:

$$g(y) = \begin{cases} y & \text{if } y \in A \\ 0 & \text{otherwise.} \end{cases}$$

Indeed, if $y \in$ range f, then $y \in A$ and $f(g(y)) = f(y) = y$.

Conversely, let A be the range of a polynomial time computable function f with a polynomial time computable inverse g. To check in polynomial time whether $y \in A$, it is enough to see whether $f(g(y)) = y$. If not, then $y \notin A$ because y is not in the range of f, by definition of an inverse. Otherwise, $y \in A$ because y is in the range of f. □

Exercise VIII.3.8 One-way functions. A polynomial time computable function with a polynomially bounded inverse but no polynomial computable inverse is called a **one-way function**.

One-way functions exist if and only if P $\neq$ NP. (Hint: if no one-way function exists, then the two previous characterizations of P and NP coincide and P = NP. Conversely, if P = NP and f has a polynomially bounded inverse, we know that

$$y \in \text{range } f \ \Rightarrow\ (\exists x)_{|x| \leq q(|y|)}(f(x) = y)$$

for some polynomial q. The range of f is in NP by VIII.3.5, and so is in P because P = NP. To define a polynomial time computable inverse g of f, let y be given. We first check if $y \in$ range f. If not, let $g(y) = 0$. Otherwise, we find successive beginnings of an x such that $f(x) = y$, as follows. Ask if $f(0) = y$. If so, stop.

Otherwise, let $x_0 = 1$. Inductively, given x_n, ask if $f(x_n) = y$. If so, stop. Otherwise, ask if

$$(\exists z)_{|x_n * 0 * z| \leq q(|y|)}(f(x_n * 0 * z) = y).$$

If so, let $x_{n+1} = 0$. Otherwise, let $x_{n+1} = 1$. The procedure stops after $q(|y|)$ steps, and is polynomial time computable because the previous question is in NP, and hence in P by hypothesis. Finally, let $g(y) = x_{q(|y|)}$.)

Berman [1977], Grollmann and Selman [1984] and Ko [1985] have proved that *one-one one-way functions exist if and only if* P $\neq$ UP, where UP is the class of sets acceptable in polynomial time by nondeterministic Turing machines with at most one accepting computation on any given input ('U' stands for 'unique acceptance').

For more on one-way functions and their connections to the theory of NP sets, see Joseph and Young [1985], Ko, Long and Du [1987], Hartmanis and Hemachandra [1987], Hemachandra [1988], Kurtz, Mahaney and Royer [1989], [1990], Selman [1992], Papadimitriou [1994].

The previous results are positive steps in the development of an analogy between the classes of r.e. and NP sets. We would like to test the analogy a bit further, along the lines of the treatment of r.e. sets in Chapters II and III. Before we do this in the next subsection, we add a positive result which, after VIII.1.18 and VIII.2.20, is unproblematic.

Theorem VIII.3.9 (Cook [1971], Karp [1972], Levin [1973a]) *There are* log-*space complete sets for* NP.

Proof. The proof of VIII.2.20 proceeds unchanged, by letting

$$\langle x, e, n \rangle \in \mathcal{K}_0^{\mathrm{NP}} \Leftrightarrow N_e \text{ accepts } x \text{ in time } |n|. \quad \square$$

Examples of real-life problems that turn out to be complete for NP are ubiquitous, and lists can be found in Garey and Jonhson [1979] (which reports around 2,000), Hopcroft and Ullman [1979], Wagner and Wechsung [1986], Johnson [1990] and Papadimitriou [1994]. Two significant examples are:

- *Boolean Satisfiability*, i.e. truth-value satisfiability of formulas of the Classical Propositional Calculus (Cook [1971], Levin [1973a])

- *Integer Linear Programming*, i.e. satisfiability over the natural numbers of systems of linear inequalities with integer coefficients (Papadimitriou [1981]).

Recall that the related problems of Horn Satisfiability and Linear Programming are instead complete for P.

Exercises VIII.3.10 Isomorphism of complete sets for NP. Recall from III.7.11 that a one-one onto function is a **recursive isomorphism** if it is recursive (automatically: together with its inverse). Similarly, a one-one onto function is a **polynomial time isomorphism** if it is polynomial time computable together with its inverse. With notation as in III.7.1, we denote by $\equiv^{\mathrm{P}}$ the equivalence relation of polynomial time isomorphism.

a) *The polynomial time m-complete sets for* NP *are all recursively isomorphic if and only if* $\mathrm{P} \neq \mathrm{NP}$. (Hint: if $\mathrm{P} = \mathrm{NP}$, then there are both finite and infinite polynomial time m-complete sets for NP, and they cannot all be recursively isomorphic. If $\mathrm{P} \neq \mathrm{NP}$, then the polynomial time m-complete sets for NP are all infinite and coinfinite recursive sets, hence recursively isomorphic by III.7.2.f and III.7.13.)

b) *If the polynomial time m-complete sets for* NP *are all polynomial time isomorphic, then* $\mathrm{P} \neq \mathrm{NP}$. (Hint: let

$$x \in A \;\Leftrightarrow\; (\exists y)(x = 2^y) \;\Leftrightarrow\; x = 10\cdots 0.$$

Then both A and $\overline{A}$ are in P, but cannot be polynomial time isomorphic for cardinality reasons.)

The open converse of part b) is called *the Hartmanis isomorphism conjecture*, and some evidence for it is provided by VIII.3.14. For a discussion and generalizations of the conjecture, see Young [1990].

Many of the known polynomial time m-complete sets for NP have indeed been proved to be polynomial time isomorphic, by applying VIII.3.11 below. See Berman and Hartmanis [1977], Hartmanis and Berman [1978], Hartmanis [1978].

Exercise VIII.3.11 Polynomial time cylinders. Recall (see VI.6.1) that a set A is a **cylinder** if it is recursively isomorphic to $A \cdot N$, where

$$A \cdot N = \{\langle x, n\rangle : x \in A\}.$$

Similarly, a set A is a **P-cylinder** if it is polynomial time isomorphic to $A \cdot N$, where $\langle x, n\rangle$ is the standard polynomial time isomorphism of $N \times N$ and N).

If A and B are P*-cylinders and $A \equiv_m^{\mathrm{P}} B$, then $A \equiv^{\mathrm{P}} B$.* (Berman and Hartmanis [1977], Mahaney and Young [1985], Dowd) (Hint: first, notice that if

$$x \in A \;\Leftrightarrow\; f_1(x) \in B$$

where f_1 is polynomial time computable, we can also suppose that f_1 has a polynomial time computable inverse. Otherwise, let h be the polynomial time isomorphism between $B \cdot N$ and B, and let

$$f_1'(x) = h(\langle f_1(x), x\rangle).$$

Then

$$x \in A \;\Leftrightarrow\; f_1(x) \in B \;\Leftrightarrow\; \langle f_1(x), x\rangle \in B \cdot N \;\Leftrightarrow\; f_1'(x) \in B.$$

Moreover, f_1' is polynomial time computable, because so are h, f_1 and the coding procedure. An inverse of f_1' is also polynomial time computable, because to obtain x

from $f'_1(x)$ one only needs to apply first an inverse of h, which is polynomial time computable by definition of polynomial time isomorphism, thus obtaining $\langle f_1(x), x\rangle$, and then take its second component by a polynomial time computable decoding function.

Similarly, with A and B interchanged. We can thus suppose that we have functions f_1 and g_1 polynomial time computable together with their inverses, such that

$$x \in A \Leftrightarrow f_1(x) \in B \qquad \text{and} \qquad x \in B \Leftrightarrow g_1(x) \in A.$$

The proof now proceeds along the lines of the Isomorphism Theorem III.7.13, illustrated in figure III.1. Let

$$f(\langle x, z\rangle) = \langle f_1(x), 2z\rangle \qquad \text{and} \qquad g(\langle x, z\rangle) = \langle g_1(x), 2z\rangle.$$

Then f and g are polynomial time computable together with their inverses, and one-one. Also, since they double the second component, an iteration of f^{-1} or g^{-1} on $\langle x, z\rangle$ stops in at most $\log z$ steps. Let

$$h(y) = \begin{cases} f(y) & \text{if } y \in R_1 \\ g^{-1}(y) & \text{if } y \in R_2, \end{cases}$$

where y is in R_1 or R_2 according to whether the process of iterating g^{-1} and f^{-1} (in this order) on y stops on the A-side or on the B-side. Then R_1 and R_2 are disjoint and exhaustive, and h is polynomial time computable. One can similarly compute h^{-1} in polynomial time, by exchanging the roles of f^{-1} and g^{-1}.)

Fenner, Kurtz and Royer [1989] have proved that *the polynomial time version of the Isomorphism Theorem III.7.13 holds if and only if* P = PSPACE. See Kurtz, Mahaney and Royer [1990] for a survey of related results.

The next notion introduces an element of novelty in the study of NP sets that is not borrowed from the study of r.e. sets.

Definition VIII.3.12 (Berman and Hartmanis [1977]) *A set A is called* **sparse** *if it has a small density, in the sense that there exists a polynomial p such that*

$$|A \cap \{0, \ldots, 2^n\}| \leq p(n).$$

Exercise VIII.3.13 *A* P*-cylinder is not sparse.* (Hint: let A be a P-cylinder, f be a one-one function computable in polynomial time q, and

$$x \in A \Leftrightarrow f(x, y) \in A.$$

Let $x \in A$ and n be given. There are 2^n elements y of length $n+1$, because the first digit must be 1. They produce elements $f(x, y)$ which are distinct by one-onenness of f, in A because x is, and of length $\leq q(n+1)$ because f is computable in time q. Thus there are at least 2^n elements in $A \cap \{0, \ldots, 2^{q(n+1)}\}$, and A is not sparse.)

Proposition VIII.3.14 (Berman and Hartmanis [1977], Berman [1978], Fortune [1979], Mahaney [1982]) *If* P $\neq$ NP*, then a polynomial time m-complete set for* NP *is not sparse.*

Proof. We prove that if there is a sparse polynomial time m-complete set A for NP, then $\mathcal{K}_0^{\rm NP} \in$ P and thus P $=$ NP. Recall that, by definition,

$$\langle x, e, n\rangle \in \mathcal{K}_0^{\rm NP} \;\Leftrightarrow\; N_e \text{ accepts } x \text{ in time } |n|.$$

To determine whether $\langle x, e, n\rangle \in \mathcal{K}_0^{\rm NP}$, we consider the tree of all computations of N_e on input x consisting of at most $|n|$ moves. Since the number of branches is exponential (see VII.4.21), we cannot directly check every branch to see whether x is accepted. The idea of the proof is to use A to reduce the number of branches we have to check to a polynomial.

Since each branch can be coded by a number, we consider the following modified version of $\mathcal{K}_0^{\rm NP}$:

$$\begin{aligned}\langle x, e, n, y\rangle \in B \quad\Leftrightarrow\quad & N_e \text{ accepts } x \text{ in time } |n|,\\ & \text{with a computation coded by a number} \leq y.\end{aligned}$$

Obviously, $B \in$ NP. Since A is polynomial time m-complete for NP, $B \leq_m^{\rm P} A$. Let f be a polynomial time computable function such that

$$\langle x, e, n, y\rangle \in B \;\Leftrightarrow\; f(x, e, n, y) \in A.$$

We use sparseness of A to cut down the number of computations we have to check, as follows. Let p be a polynomial such that A contains at most $p(n)$ elements of length n, and let q be a polynomial such that f is computable in time q. Since $|f(x, e, n, y)| \leq q(|x|, |e|, |n|, |y|)$, it follows that A contains at most $p(q(|x|, |e|, |n|, |y|))$ elements of the form $f(x, e, n, y)$.

First, given x, e and n, we compute in polynomial time a bound Y of all codes of possible computations of N_e on x taking at most $|n|$ steps (the order of magnitude of Y is $c^{|n|}$, which is polynomial time computable by VIII.2.4). Second, we let $m = p(q(|x|, |e|, |n|, |Y|)) + 1$ and evenly divide the numbers up to Y into $m + 1$ intervals of equal length, determined by numbers y_i such that

$$0 < y_1 < \cdots < y_m < Y.$$

Third, we consider the following two possibilities:

- *for some* $0 < i < j < m$, $f(x, e, n, y_i) = f(x, e, n, y_j)$
  Then
$$f(x, e, n, y_i) \in A \;\Leftrightarrow\; f(x, e, n, y_j) \in A,$$

and hence

$$\langle x, e, n, y_i \rangle \in B \;\Leftrightarrow\; \langle x, e, n, y_j \rangle \in B,$$

i.e. N_e accepts x in time $|n|$ with a computation coded by a number $\leq y_j$ if and only if N_e accepts x in time $|n|$ with a computation coded by a number $\leq y_i$.

This means that we can eliminate all computations coded by numbers y such that $y_i < y \leq y_j$.

- *for all* $0 < i, j < m$, $f(x, e, n, y_i) \neq f(x, e, n, y_j)$
Then, by sparseness of A and the choice of m, at least one $f(x, e, n, y_i)$ is not in A. This means that N_e does not accept x in time $|n|$ with computations coded by numbers $\leq y_i$. Since $y_1 \leq y_i$, we can thus eliminate all computations coded by numbers $\leq y_1$.

In both cases we are thus able to eliminate a good number of possible computations, and to concentrate on the remaining ones. By repeating the procedure at most a polynomial number of times, we eventually reduce the exponential number of computations to be checked to a polynomial number, because each time an interval of at most exponentially many numbers is evenly divided into polynomially many subintervals. We can then check directly the remaining polynomial number of computations in polynomial time, to see whether any of them accepts x. □

The last result can be seen as a first step in the direction of proving that if P $\neq$ NP, then all polynomial time m-complete sets for NP are polynomial time isomorphic, since it shows that at least they have the same high density.

Other results on sparse NP sets are proved in the rest of the chapter, see VIII.3.15.b, VIII.4.15.b, VIII.4.16.c, and VIII.6.8.

More results on sparse sets are surveyed in Mahaney [1986], [1989] and Young [1990], [1992] and proved in Lynch [1975a], Solovay [1976], Hartmanis [1978], Karp and Lipton [1980], Long [1982], Mahaney [1982], Ukkonen [1983], Sewelson [1983], Yap [1983], Yesha [1983], Hartmanis and Yesha [1984], Long and Selman [1986], Balcázar, Book and Schöning [1986], [1986a], Schöning [1986], Kadin [1987], [1988], Ko [1989], Ogiwara and Watanabe [1991], Hemachandra, Ogiwara and Watanabe [1992], Homer and Longpré [1994].

In particular, Ogiwara and Watanabe [1991] prove that *if* P $\neq$ NP*, then a polynomial time btt-complete set for* NP *is not sparse* (the proof of VIII.3.14 given above is a special case of their proof).

Exercises VIII.3.15 Census function. Given a set A, the **census function** of A is defined as follows:

$$C_A(n) = |A \cap \{0, \ldots, 2^n\}|.$$

a) *If A is an r.e. set and C_A is recursive, then $\overline{A}$ is also r.e. (and so A is recursive).* (Hint: given x of length n, generate A until $C_A(n)$ elements of length $\leq n$ have been found, and see if x is among them.)

b) *If A is a sparse set in* NP *and C_A is polynomial time computable, then $\overline{A}$ is also in* NP. (Hartmanis and Mahaney [1980]) (Hint: given x of length n, guess $C_A(n)$ elements of length $\leq n$, check whether they are all in A, and if so see if x is among them. The procedure is in NP because so is A, and C_A is bounded by a polynomial because A is sparse.)

NP sets as analogues of r.e. sets: failures ⋆

The next major result proved in Section II.1 on r.e. sets (after II.1.10 and II.1.16, of which VIII.3.3 and VIII.3.5 are the analogues) was Post's Theorem II.1.19, which stated that if both A and $\overline{A}$ are r.e., then A is recursive (the converse being obvious). The proof was simply the following: to decide whether y is in A, run recursive enumerations of A and $\overline{A}$ in parallel, until y is enumerated by one of them.

The analogue of this result would be that if both A and $\overline{A}$ are in NP, then A is in P. Since the converse is again obvious, this can be succinctly stated as

$$\mathrm{P} = \mathrm{NP} \cap \text{co-NP},$$

and is a major open problem. Notice that an analogue of the simple proof given above fails for the following reason: when running polynomial time enumerations (with polynomially bounded inverses) of A and $\overline{A}$ in parallel, a given y is indeed enumerated in one of them, but not necessarily quickly. More specifically, if f has a polynomially bounded inverse, then we do know that any $y \in$ range f is enumerated at a stage x such that $|x| \leq q(|y|)$, for some polynomial q, but this only gives the bound $x \leq 2^{q(|y|)}$, while we would need a polynomial in $|y|$.

A similar problem appears for II.1.20, which stated that an infinite r.e. set has an infinite recursive subset. The analogue, that

$$\text{every infinite NP set has an infinite P subset,}$$

is again an open problem.

The next result about r.e. sets was the crucial II.2.3, which showed the existence of an r.e. nonrecursive set. The analogue can be succinctly stated as

$$\mathrm{P} \neq \mathrm{NP}.$$

This is obviously *the most fundamental open problem in the theory of* NP *sets*. The first explicit statement of the problem appears in a 1956 letter from Gödel to von Neumann, as a conjecture that Boolean Satisfiability is polynomial time computable (see Sipser [1992]). However, an implicit recognition of the significance of the problem is much older, as witnessed by the standard use of Euclid's algorithm for the greatest common divisor instead of the obvious algorithm requiring prime decomposition (the first algorithm was proved to be polynomial time computable by Lamé [1884], while the second is only known to be computable in nondeterministic polynomial time). For the history of the problem in Computer Science, see Trakhtenbrot [1984] and Sipser [1992].

Recall that II.2.3 is proved by defining a uniform r.e. enumeration $\{\mathcal{W}_e\}_{e\in\omega}$ of the r.e. sets, and diagonalizing over it by letting

$$x \in \mathcal{K} \;\Leftrightarrow\; x \in \mathcal{W}_x.$$

Then

$$\mathcal{W}_x \subseteq \overline{\mathcal{K}} \;\Rightarrow\; x \in \overline{\mathcal{K}} - \mathcal{W}_x,$$

and thus $\overline{\mathcal{K}}$ and $\mathcal{K}$ were, respectively, productive and creative (III.6.5). In particular, $\overline{\mathcal{K}}$ was not r.e., and thus $\mathcal{K}$ could not be recursive.

Since this proof uses the fact that a recursive set is r.e. together with its complement, which holds in the present analogue, it might appear that there is hope for a similar approach. We now show that this is not the case, because of other obstacles.

The next definition is a weak analogue of II.5.2 and imposes only a trivial restriction on enumerations of NP.

Definition VIII.3.16 *An enumeration $\{\mathcal{W}_e^{\mathrm{NP}}\}_{e\in\omega}$ of* NP *is* **acceptable** *if, for any deterministic Turing machine M, there is a polynomial time computable function h such that $\mathcal{W}_{h(e)}^{\mathrm{NP}}$ is the set in* NP *computed by M in time $|x|^e$.*

For the moment, we fix an acceptable enumeration of NP. The next definition is an analogue of III.6.5.

Definition VIII.3.17 *A set A is* **P-productive** *if there is a polynomial time computable function f such that, for every x,*

$$\mathcal{W}_x^{\mathrm{NP}} \subseteq A \;\Rightarrow\; f(x) \in A - \mathcal{W}_x^{\mathrm{NP}}.$$

A set A is **NP-creative** *if it is in* NP *and co*-P*-productive.*

Proposition VIII.3.18 (Ko and Moore [1981]) *A* P*-productive set is not computable in deterministic time $2^{p(|x|)}$, for any polynomial p.*

Proof. Suppose A is P-productive and computable by a deterministic Turing machine M in time $2^{p(|x|)}$, for some polynomial p. Let f be a function computable in time $q(|x|)$, for some polynomial q, and such that

$$\mathcal{W}_x^{\mathrm{NP}} \subseteq A \;\Rightarrow\; f(x) \in A - \mathcal{W}_x^{\mathrm{NP}}.$$

Let h be a function computable in time $r(|x|)$, for some polynomial r, and such that $\mathcal{W}^{\text{NP}}_{h(e)}$ is the NP set computed by M in time $|x|^e$.

By construction, $\mathcal{W}^{\text{NP}}_{h(e)} \subseteq A$. Thus $f(h(e)) \in A - \mathcal{W}^{\text{NP}}_{h(e)}$. Since $f(h(e)) \in A$, M accepts $f(h(e))$ in time $2^{p(|f(h(e))|)}$. But

$$p(|f(h(e))|) \leq p(q(|h(e)|)) \leq p(q(r(|e|))),$$

because f and h are computable in time q and r, and thus $|f(x)|$ and $|h(x)|$ are bounded by $q(|x|)$ and $r(|x|)$.

Choose e large enough so that $p(q(r(|e|))) \leq 2^{|e|-1} \leq e$. Then M accepts $f(h(e))$ in time 2^e, and hence in time $|f(h(e))|^e$. Then, by definition of $\mathcal{W}^{\text{NP}}_{h(e)}$, $f(h(e)) \in \mathcal{W}^{\text{NP}}_{h(e)}$, which is a contradiction. □

In the notation introduced in VIII.6.10, the previous result says that a P-productive set is not in POLYEXP.

Corollary VIII.3.19 *There is no* NP*-creative set.*

Proof. By VII.4.21, every NP set is computable in deterministic time $2^{p(|x|)}$, for some polynomial p. Hence so is any co-NP set, since deterministic classes are closed under complements. Thus no co-NP set can be P-productive, and no NP set can be NP-creative. □

Corollary VIII.3.20 *No enumeration of* NP *can be both acceptable and uniform.*

Proof. Otherwise, the set defined by

$$x \in \mathcal{K}^{\text{NP}} \Leftrightarrow x \in \mathcal{W}^{\text{NP}}_x$$

would be in NP by the uniformity of the enumeration. In particular, $\mathcal{K}^{\text{NP}}$ would be NP-creative. □

It follows from the Hierarchy Theorem for NP of Cook [1973] that *there is no uniform enumeration of* NP. Otherwise, the set

$$\langle x, e\rangle \in A \Leftrightarrow x \in \mathcal{W}^{\text{NP}}_e$$

would be in NP_n for some n, and hence

$$\text{NP} \subseteq \text{NP}_n \subset \text{NP}_{n+1} \subseteq \text{NP},$$

which is a contradiction.

Exercises VIII.3.21 a) *For each $n \geq 1$ there is a coinfinite set $A_n \in$ NP and a polynomial time computable function f such that, for every* NP *set $\mathcal{W}_e^{\text{NP}}$ accepted in time $|x|^n$,*

$$\mathcal{W}_e^{\text{NP}} \subseteq \overline{A}_n \ \Rightarrow\ f(e) \in \overline{A}_n - \mathcal{W}_e^{\text{NP}}.$$

(Joseph and Young [1985]) (Hint: let

$$x \in A_n \ \Leftrightarrow\ \text{some computation of } N_x \text{ halts in time } |x|^n.)$$

b) *There is a* P*-productive set computable in deterministic time $2^{2^{c\cdot|x|}}$, for some constant c.* (Ko and Moore [1981]) (Hint: let

$$x \in A \ \Leftrightarrow\ x \in A_x,$$

where A_x is the set defined in part a).)

Part a) provides a restricted analogue of productiveness. See Wang [1989] for other analogues, in particular of complete productiveness III.6.1.

The last results bring us into the realm of properties of r.e. sets investigated in Chapter III. Without indulging in a premature development of an area whose existence is still in doubt, we limit ourselves to analogues of III.2.9 and III.4.13.

Definition VIII.3.22 *A set A is* **NP-simple** *if it is in* NP*, and its complement is infinite and does contain infinite* NP *sets.*

A set A is **NP-maximal** *if it is in* NP*, and its complement cannot be split into two infinite parts by an* NP *set.*

The existence of NP-simple sets is an open problem, and a positive answer would imply P $\neq$ NP (since the complement of a NP-simple set is not in NP, while every set in P has an NP complement).

Exercise VIII.3.23 *For each n there is a coinfinite set $A_n \in$ NP such that $\overline{A}_n$ does not contain infinite* NP *sets accepted in time $|x|^n$.* (Homer [1986]) (Hint: let $\{V_e\}_{e\in\omega}$ be a recursive enumeration of the NP sets accepted in time $|x|^n$, for a fixed n. For each e, we want to satisfy the requirement

$$R_e\ :\ V_e \text{ infinite} \ \Rightarrow\ V_e \cap A \neq \emptyset.$$

At stage x we decide whether x is in A_n, as follows. We simulate the previous construction for $|x|$ steps, and determine which requirements have been satisfied, as well as the number m_x of elements that have entered $\overline{A}_n$ by then. We put x into A_n if and only if there is e such that

$$|e| \leq |x| \ \wedge\ e < m_x \ \wedge\ R_e \text{ not yet satisfied} \ \wedge\ x \in V_e.$$

Then $A_n \in$ NP because V_e is computable in time $|x|^n$ for a fixed n, independently of e. Moreover, $\overline{A}_n$ is infinite because R_e can put a new element in A_n only if there are already at least e elements in $\overline{A}_n$, by the condition on m_x.)

The next result takes care of the second notion introduced in VIII.3.22, and is interesting on its own.

Proposition VIII.3.24 (Breidbart [1978]) *Every infinite recursive set can be split into two infinite parts by a set in* P.

Proof. Let A be an infinite recursive set. Given a partial recursive function φ_e, we define a 0,1-valued polynomial time computable function $\varphi_{f(e)}$ uniformly in e, as follows. Given x, we dovetail enumerations of

$$A \cap \{x : \varphi_e(x) \simeq 1\} \qquad \text{and} \qquad A \cap \{x : \varphi_e(x) \simeq 0\}$$

for $|x|$ steps. We then output 1 if less elements are generated in the former, and 0 otherwise.

By the Fixed Point Theorem, there is e such that $\varphi_{f(e)} \simeq \varphi_e$. By definition of $\varphi_{f(e)}$, φ_e is a polynomial time computable characteristic function, and hence it defines a set $B \in$ P.

It remains to prove that both $A \cap B$ and $A \cap \overline{B}$ are infinite. Suppose that, for example, $A \cap B$ is finite. Since A is infinite, so is $A \cap \overline{B}$. Then there is a stage after which, in the dovetailed enumeration of $A \cap B$ and $A \cap \overline{B}$, less elements have always been generated in the former. From that stage on the procedure always outputs 1, and thus B is cofinite. Since A is infinite, $A \cap B$ is then infinite, which is a contradiction.

The case in which $A \cap \overline{B}$ is finite is symmetric. $\square$

Corollary VIII.3.25 *There is no* NP*-maximal set.*

Notice that the proof actually shows also that, with obvious terminology, *there is no* NP*-r-maximal set* (see IX.2.12 for the definition of an r-maximal set).

Corollary VIII.3.26 *The theories of r.e. and* NP *sets modulo finite sets, ordered by inclusion, are not elementarily equivalent.*

Proof. The notion of a maximal set can easily be defined in the lattice of r.e. sets modulo finite sets, as a maximal element (see p. I.288). $\square$

We do not know whether the same result holds for the theories of r.e. and NP sets, ordered by inclusion, since the definability of finiteness in the lattice of r.e. sets (IX.3.1) uses properties of r.e. sets not known to hold for NP sets.

Relativizations

After introducing the notion of relative recursiveness in II.3.5, we have noticed that a number of results (although, by V.7.13, not all) relativize to arbitrary oracles. For example, for any oracle A, there are sets r.e. in A and not recursive in A. We now deal with relativizations in the context of polynomial time computability.

Definition VIII.3.27 $\mathbf{P}^{A}$ *is the class of sets computable in polynomial time (in the length of the input) by a deterministic Turing machine with oracle A.*

$\mathbf{NP}^{A}$ *is the class of sets acceptable in polynomial time (in the length of the input) by a nondeterministic Turing machine with oracle A.*

In terms of polynomial time Turing reducibility (see VIII.2.29), one has

$$\mathrm{P}^{A} = \{B : B \leq_{T}^{\mathrm{P}} A\}$$

and thus, in the terminology of Chapter V, P^{A} is the principal ideal generated by A.

Obviously, by definition, $\mathrm{P}^{A} \subseteq \mathrm{NP}^{A}$. The analogue of the relativized result about r.e. sets quoted above now becomes $\mathrm{P}^{A} \neq \mathrm{NP}^{A}$. We can prove that *the answer to the problem* $\mathrm{P} = \mathrm{NP}$ *does not relativize* by showing that there are oracles for which such an answer is positive, and oracles for which it is negative.

Proposition VIII.3.28 (Baker, Gill and Solovay [1975], Meyer, Fischer, Hunt) *There is a (recursive) oracle A such that* $\mathrm{P}^{A} = \mathrm{NP}^{A}$.

Proof. We only have to ensure $\mathrm{NP}^{A} \subseteq \mathrm{P}^{A}$, since the converse holds automatically. The intuitive idea is to let A be a polynomial time T-complete set for NP^{A}, so that every set in NP^{A} is polynomial time T-reducible to A, i.e. in P^{A}.

By relativization of VIII.3.9, for any oracle A a polynomial time T-complete set for NP^{A} is defined by

$$\langle x, e, n\rangle \in \mathcal{K}_0^{\mathrm{NP}^{A}} \Leftrightarrow N_e^{A} \text{ accepts } x \text{ in time } |n|.$$

To have $A = \mathcal{K}_0^{\mathrm{NP}^{A}}$, we let

$$\langle x, e, n\rangle \in A \Leftrightarrow N_e^{A} \text{ accepts } x \text{ in time } |n|.$$

It only remains to show that A is well-defined. We proceed by induction on the length of the input. To see whether $\langle x, e, n\rangle$ is in A, we have to run N_e^{A} for at most $|n|$ steps, and thus no query to the oracle can be made for numbers of length $> |n|$. Since $|n| < |\langle x, e, n\rangle|$ by definition of coding, the oracle is thus queried only for numbers less than $\langle x, e, n\rangle$, on which A has already been defined (by induction hypothesis). $\square$

It is not known whether it is enough to let A be a polynomial time T-complete set for NP, to have $\mathrm{P}^A = \mathrm{NP}^A$.

Theorem VIII.3.29 (Dekhtiar [1969], Baker, Gill and Solovay [1975], Ladner) *There is a (recursive) oracle B such that $\mathrm{P}^B \neq \mathrm{NP}^B$.*

Proof. We can only ensure that $\mathrm{NP}^B \subseteq \mathrm{P}^B$ fails, since the converse holds automatically. Thus we build $A \in \mathrm{NP}^B - \mathrm{P}^B$. The intuitive idea is to let membership of a single element in A be determined by exponentially many choices on B. Since a polynomial time computation cannot use all the possibilities, at least one remains that allows us to diagonalize against any such a computation, so that A is not in P^B. On the other hand, a single nondeterministic guess is enough, and A is in NP^B.

To obtain $A \in \mathrm{NP}^B$, by VIII.3.3 we can let

$$\begin{aligned} x \in A \quad &\Leftrightarrow \quad (\exists y)(2^{|x|-1} \leq y < 2^{|x|} \ \wedge \ y \in B) \\ &\Leftrightarrow \quad (\exists y)(|y| = |x| \ \wedge \ y \in B). \end{aligned}$$

To obtain $A \notin \mathrm{P}^B$, we diagonalize. Let $\{p_n\}_{n\in\omega}$ be a recursive enumeration of the polynomials in one variable. The requirements are:

$$R_{\langle e,n\rangle} \ : \ A \not\simeq \{e\}^B \text{ with time bound } p_n.$$

We build B by finite initial segments σ_s (see Section V.2). We start with $\sigma_0 = \emptyset$. At stage $s+1$, let σ_s be given. If $s = \langle e, n\rangle$, then we attack $R_{\langle e,n\rangle}$. We choose a number l such that:

- $l+1$ is greater than the length of any element on which σ_s is defined, i.e.

$$l + 1 > \max\{|z| : \sigma_s(z)\!\downarrow\}$$

- 2^l is greater than $p_n(l+1)$.

Let $x = 2^l$, so that $|x| = l+1$ (because, when written in binary, x consists of a 1 followed by l 0's). Compute $\{e\}^{\sigma_s}_{p_n(|x|)}(x)$, and let $\sigma \supseteq \sigma_s$ be the extension of σ_s defined by letting $\sigma(z) = 0$ for all elements z used negatively in the computation (i.e. such that z is not in the finite set defined by σ_s, and the oracle has been queried about z). By letting $B \supseteq \sigma$, we ensure that

$$\{e\}^B_{p_n(|x|)}(x) \simeq \{e\}^{\sigma_s}_{p_n(|x|)}(x).$$

There are three cases:

- $\{e\}^B_{p_n(|x|)}(x)\uparrow$
  Then $\{e\}^B(x)$ does not converge in $p_n(|x|)$ steps, $R_{\langle e,n\rangle}$ is vacuously satisfied, and we need only let σ_{s+1} be the smallest initial segment extending σ.

- $\{e\}^B_{p_n(|x|)}(x)\downarrow$ *but* $\{e\}^B_{p_n(|x|)}(x) \not\simeq 0$
  Then we want $x \in \overline{A}$, and for this we have to put all elements of length $|x| = l+1$ into $\overline{B}$. Since, by the choice of l, σ_s was not defined on any element of such a length, and σ did not assign 1 to any new element, we can let σ_{s+1} be the smallest initial segment extending σ and such that $\sigma_{s+1}(z) = 0$ for all z of length $l+1$.

- $\{e\}^B_{p_n(|x|)}(x) \simeq 0$
  Then we want $x \in A$, and for this we have to put some element of length $|x| = l+1$ into B. Since, by the choice of l, σ_s was not defined on any element of such a length, and σ has been defined for at most $p_n(l+1) < 2^l$ new elements, among the 2^l elements of length $l+1$ there is one z such that $\sigma(z)\uparrow$. Let σ_{s+1} be the smallest initial segment extending σ and such that $\sigma_{s+1}(z) = 1$.

Since the construction is recursive, so is B. $\square$

The oracle B is constructed by the finite extension method. Thus, by V.3.13, the set $\{B : \mathrm{P}^B \neq \mathrm{NP}^B\}$ is comeager (Bennett and Gill [1981] have shown that it also has measure 1). In particular, the finite extension method cannot be used to prove VIII.3.28.

The results proved above show that the problem P = NP cannot be settled by methods that relativize. However, this restriction is purely methodological and it does not say anything about the difficulty of the problem. To show this, we exhibit an example of *a provable statement that admits conflicting relativizations, and such that most relativizations contradict the unrelativized statement* (Hartmanis [1985]). Notice that the proof of the two results above relativize, in the sense that for any oracle C there are oracles A and B such that

$$(\mathrm{P}^C)^A = (\mathrm{NP}^C)^A \qquad \text{and} \qquad (\mathrm{P}^C)^B \neq (\mathrm{NP}^C)^B.$$

By taking as C the oracle constructed in VIII.3.29, we thus have an example of a provable assertion (i.e. $\mathrm{P}^C = \mathrm{NP}^C$) that does not relativize. Since the set $\{B : (\mathrm{P}^C)^B \neq (\mathrm{NP}^C)^B\}$ is comeager, most relativizations contradict the unrelativized result (i.e. $\mathrm{P}^C = \mathrm{NP}^C$).

The previous example is somewhat artificial, but Hartmanis, Chang, Chari, Ranjan and Rohatgi [1992] provide natural examples from Theoretical Computer Science of provable statements with conflicting relativizations.

Exercises VIII.3.30 a) $\{B : B$ *recursive* $\wedge$ $\mathrm{P}^B \neq \mathrm{NP}^B\}$ *is* Π^0_2*-complete in the characteristic indices of recursive sets.* (Hájek [1979]) (Hint: let $\{C_e\}_{e\in\omega}$ be an enumeration of the recursive sets by characteristic indices, see II.5.9. To show that the set is Π^0_2, notice that

$$\mathrm{P}^{C_e} \neq \mathrm{NP}^{C_e} \;\Leftrightarrow\; \mathcal{K}_0^{\mathrm{NP}^{C_e}} \notin \mathrm{P}^{C_e},$$

where $\mathcal{K}_0^{\mathrm{NP}^{C_e}}$ is the canonical complete set for NP^{C_e}.

To show Π^0_2-completeness, use X.9.6. Fix a recursive set C such that $\mathrm{P}^C = \mathrm{NP}^C$, and modify VIII.3.29 to define a recursive function f such that

$$\mathcal{W}_e \text{ infinite} \;\Leftrightarrow\; \mathrm{P}^{C_{f(e)}} \neq \mathrm{NP}^{C_{f(e)}}$$

by ensuring that $C_{f(e)}$ differs finitely from C when $\mathcal{W}_e$ is finite.)

b) *There are recursive oracles A and B such that* $\mathrm{P}^A = \mathrm{NP}^A$ *and* $\mathrm{P}^B \neq \mathrm{NP}^B$*, but both assertions are independent from ZFC.* (Hartmanis and Hopcroft [1976]) (Hint: this follows from part a), since both

$$\{e : \mathrm{P}^{C_e} = \mathrm{NP}^{C_e} \text{ is provable in } ZFC\}$$

and

$$\{e : \mathrm{P}^{C_e} \neq \mathrm{NP}^{C_e} \text{ is provable in } ZFC\}$$

are Σ^0_1.)

Kintala and Fischer [1980], and Xu, Doner and Book [1983] have proved the following *refinements* of the results above, where $\mathrm{NP}[g(x)]$ is the class of sets acceptable in nondeterministic polynomial time, by making at most $g(x)$ nondeterministic moves on input x.

- $\mathrm{NP}[\log|x|] = \mathrm{P}$, and hence *little nondeterminism is avoidable*. The reason is that a nondeterministic Turing machine with at most c instructions, working on input x in polynomial time $p(|x|)$, and making at most $\log|x|$ nondeterministic moves can only make $c^{\log|x|} = |x|^{\log c}$ different computations, each taking at most time $p(|x|)$. Thus in time $|x|^{\log c}\cdot p(|x|)$ one can simulate all possible computations.

- There is an oracle B such that

  $$\mathrm{P}^B \subset \mathrm{NP}^B[(\log|x|)^2] = \mathrm{NP}^B,$$

  and hence *polynomial nondeterminism might be unavoidable, but reducible to small amounts*.

- There is an oracle B such that

  $$\mathrm{NP}^B[|x|] \subset \mathrm{NP}^B[|x|^2] \subset \cdots \subset \mathrm{NP}^B,$$

  and hence *no bound might exist on the amount of unavoidable polynomial nondeterminism*.

Various ways have been found to avoid the phenomenon of conflicting relativizations, and produce so-called *positive relativizations*. Usually one restricts either the access to the oracle in a computation (see VIII.5.18), or the class of oracles (see VIII.4.14 and VIII.5.20). These results are interesting not because they 'solve' the problem of conflicting relativizations, but rather because they isolate properties used in the proofs of the conflicting standard relativizations, thus clarifying the phenomenon.

For the collapse problem considered in this subsection, Selman, Xu and Book [1983], Book, Long and Selman [1984], and Long [1985] have defined NP_b^A (where 'b' stands for *bounded number of queries*) as the class of sets accepted by nondeterministic Turing machines working in polynomial time, and such that the number of queries in the tree of all possible computations on a given input is polynomially bounded (without this restriction, a polynomial number $p(|x|)$ of queries can be made for each one of $c^{p(|x|)}$ computations, and thus the total number of queries is bounded only by $2^{2^{|x|}}$). Then

$$\mathrm{P} = \mathrm{NP} \;\Leftrightarrow\; (\forall A)(\mathrm{P}^A = \mathrm{NP}_b^A).$$

One possible way of meeting the condition just stated (first considered by Morris) is to consider *oracle tape machines*, in which the oracle is coded on a special tape (e.g. in increasing order), and a query to the oracle is answered by (serially) searching for the information about a queried element on the oracle tape. A similar result is proved in VIII.5.18. For an exposition of other results about relativization of $\mathrm{P} = \mathrm{NP}$ see Book [1989], and Balcázar, Díaz and Gabarró [1990].

We turn now to relativizations of the problems

$$\mathrm{P} = \mathrm{NP} \cap \text{co-NP} \qquad \text{and} \qquad \mathrm{NP} = \text{co-NP}.$$

Since P^A is closed under complements, VIII.3.28 already provides a relativization for which these two equalities (simultaneously) hold. We thus turn to relativizations for which they fail.

Proposition VIII.3.31 (Ladner, Fischer, Meyer) *There is a (recursive) oracle B such that*

$$\mathrm{P}^B \neq \mathrm{NP}^B \qquad \textit{and} \qquad \mathrm{NP}^B = \textit{co}\text{-}\mathrm{NP}^B,$$

and hence such that

$$\mathrm{P}^B \neq \mathrm{NP}^B \cap \textit{co}\text{-}\mathrm{NP}^B.$$

Proof. To make $\mathrm{P}^B \neq \mathrm{NP}^B$, we build $A \in \mathrm{NP}^B - \mathrm{P}^B$. To make $\mathrm{NP}^B = \text{co-NP}^B$ we ensure that $\mathcal{K}_0^{\mathrm{NP}^B} \in \text{co-NP}^B$, so that $\mathrm{NP}^B \subseteq \text{co-NP}^B$. Then, symmetrically, $\text{co-NP}^B \subseteq \mathrm{NP}^B$, and thus $\mathrm{NP}^B = \text{co-NP}^B$.

To obtain $A \in \mathrm{NP}^B - \mathrm{P}^B$, as in VIII.3.29, we let

$$x \in A \;\Leftrightarrow\; (\exists y)(|y| = |x| \,\wedge\, y \in B)$$

and satisfy the requirements

$$R_{\langle e,n \rangle} \;:\; A \not\simeq \{e\}^B \text{ with bound } p_n,$$

where $\{p_n\}_{n \in \omega}$ is a recursive enumeration of the polynomials in one variable.

To obtain $\mathcal{K}_0^{\mathrm{NP}^B} \in$ co-NP^B, we build B in such a way that

$$x \notin \mathcal{K}_0^{\mathrm{NP}^B} \quad\Leftrightarrow\quad (\exists y)(|y| = |x| \,\wedge\, x * y \in B),$$

where $x * y$ is the number whose binary representation is the concatenation of the binary representations of x and y. Thus we use only elements of length $2 \cdot |x|$ (in particular, of even length) to code whether x is in $\mathcal{K}_0^{\mathrm{NP}^B}$. This will leave the elements of odd length free for diagonalization purposes.

As in VIII.3.28, using $\mathcal{K}_0^{\mathrm{NP}^B}$ in the definition of B is possible because, inductively, membership of an element into $\mathcal{K}_0^{\mathrm{NP}^B}$ requires only a definition of B for elements of smaller length. In particular, the construction will have to ensure that when membership of elements of even length is decided, membership of all elements of smaller length has already been decided.

Finally, to be able to put an element of even length in B when needed, at least one should be available (i.e. its membership is not yet decided). For any requirement $R_{\langle e,n \rangle}$, the diagonalization procedure of VIII.3.29 puts at most one element in B, with the same length as a chosen witness. By choosing a witness of odd length, we can avoid interference with the rest of the construction. But the same procedure restrains a number of elements from entering B, and these we cannot choose (because they depend on a computation on the chosen witness). Without restraint we might define B on too many elements, and have exhausted all elements of a given even length by the time we need one. The simple strategy of VIII.3.29, of satisfying one requirement at any given stage, must thus be replaced by the strategy of satisfying requirements when the chance arises, i.e. by a cancellation argument in the style of Chapter VII (see p. 29).

We build B by finite strings σ_s. We start with $\sigma_0 = \emptyset$. At stage $s + 1$, let σ_s be given. By induction, σ_s is defined on all elements of length $\leq s$, and it has value 0 on the elements of length $> s$ on which it is defined.

1. *if $s = 2m$, then we work for* $\mathrm{P}^B \neq \mathrm{NP}^B$
   Let $x = 2^s$, so that $|x| = s + 1$ is odd. Consider the $\langle e, n \rangle$ such that:
   - $\langle e, n \rangle \leq s$

- $\langle e, n\rangle$ has not yet been cancelled
- σ_s is not defined on any element of length $s+1$
- $p_n(s+1) < 2^m$.

If there is no such $\langle e, n\rangle$, let σ_{s+1} be the extension of σ_s such that $\sigma_{s+1}(z) = 0$ for all elements z of length $s+1$.

If there is such an $\langle e, n\rangle$, choose the least one $\langle e_s, n_s\rangle$, cancel it, and proceed as in VIII.3.28 (where now $l = s$), with the only added condition of defining σ_{s+1} for all elements of length $s+1$.

More precisely, if σ is the extension of σ_s defined by letting $\sigma(z) = 0$ for all elements used negatively in the computation of $\{e_s\}^{\sigma_s}_{p_{n_s}(|x|)}(x)$, then:

- if the computation does not converge to 0, let σ_{s+1} be the extension of σ obtained by letting $\sigma_{s+1}(z) = 0$ for all z of length $s+1$
- if the computation converges to 0, let σ_{s+1} be the extension of σ obtained by letting $\sigma_{s+1}(z) = 0$ for all z of length $s+1$ but one, and $\sigma_{s+1}(z) = 1$ for one such z.

Notice that all indices are eventually cancelled. Indeed, the second part of the construction only affects numbers of even length, and thus by not restraining computations long enough, we do get to odd stages $s+1$ such that σ_s is not defined for any element of length $s+1$.

2. *if $s = 2m+1$, then we work for* $\mathrm{NP}^B = co$-NP^B
As in VIII.3.28, for each x of length $m+1 = \frac{s+1}{2}$,

$$x \in \mathcal{K}_0^{\mathrm{NP}^{\sigma_s}} \Leftrightarrow x \in \mathcal{K}_0^{\mathrm{NP}^B},$$

because σ_s has already been defined on all elements of length $\leq |x|-1 \leq s$.

For each x of length $m+1$, we need y_x of length $m+1$ such that σ_s has not yet been defined on $x * y_x$, so that we can let σ_{s+1} be the extension of σ_s such that

$$\sigma_{s+1}(x * y_x) = \begin{cases} 1 & \text{if } x \notin \mathcal{K}_0^{\mathrm{NP}^{\sigma_s}} \\ 0 & \text{otherwise,} \end{cases}$$

thereby ensuring

$$x * y_x \in B \Leftrightarrow x \notin \mathcal{K}_0^{\mathrm{NP}^{\sigma_s}},$$

and such that $\sigma_{s+1}(x * y) = 0$ on the remaining $y \neq y_x$ of length $m+1$.

We only have to argue that y_x exists. But the only stages that could have involved elements of length $s+1$ are those of the form $2n+1$ with

$n \le m$, each of which could restrain at most 2^n elements. Since there are 2^{m+1} strings y of 0's and 1's of length $m+1$, and hence numbers $x * y$, and only

$$(\sum_{n \le m} 2^n) < 2^{m+1}$$

could have been restrained, at least one is available.

As usual, the oracle B is recursive because such is the construction. □

A number of refinements, some of which are considered in the exercises, are possible. In particular, *the hypothesis* $\mathrm{P} \neq \mathrm{NP}$ *does not settle the problems*

$$\mathrm{P} = \mathrm{NP} \cap \textit{co}\text{-NP} \qquad \textit{and} \qquad \mathrm{NP} = \textit{co}\text{-NP}$$

in a way that relativizes. Moreover, by considering equalities and strict inclusions, *all four possibilities in*

$$\mathrm{P}^B \subseteq \mathrm{NP}^B \cap \textit{co}\text{-NP}^B \subseteq \mathrm{NP}^B$$

are realized (Baker, Gill and Solovay [1975]).

Exercises VIII.3.32 a) *If* $\mathrm{NP}^B \neq \textit{co}\text{-NP}^B$*, then* $\mathrm{P}^B \neq \mathrm{NP}^B$.

b) *There is an oracle* B *such that* $\mathrm{NP}^B \neq \textit{co}\text{-NP}^B$. (Baker, Gill and Solovay [1975]) (Hint: let

$$x \in A \Leftrightarrow (\exists y)(|y| = |x| \wedge y \in B),$$

and ensure that $\overline{A} \notin \mathrm{NP}^B$, and hence that $A \notin \text{co-NP}^B$, as in VIII.3.29, by diagonalizing against nondeterministic machines working in polynomial time.)

c) *There is an oracle* B *such that* $\mathrm{NP}^B \neq \textit{co}\text{-NP}^B$*, but the assertion is independent from ZFC*. (Grant [1980]) (Hint: see VIII.3.30.)

d) *There is an oracle* B *such that* $\mathrm{P}^B \subset \mathrm{NP}^B \cap \textit{co}\text{-NP}^B \subset \mathrm{NP}^B$. (Baker, Gill and Solovay [1975]) (Hint: this follows from VIII.3.33.d. For a direct proof, let

$$\begin{aligned} x \in A \quad &\Leftrightarrow \quad |x| \text{ even} \wedge (\exists y)(|y| = |x| \wedge y \in B) \\ x \in C \quad &\Leftrightarrow \quad |x| \text{ odd} \wedge (\exists y)(|y| = |x| - 2 \wedge 11 * y \in B). \end{aligned}$$

Then $A, C \in \mathrm{NP}^B$. We ensure as usual that $\overline{A} \notin \mathrm{NP}^B$ and $C \notin \mathrm{P}^B$. To also obtain $\overline{C} \in \mathrm{NP}^B$, we require that

$$(\exists y)(|y| = n \wedge 11 * y \in C) \Leftrightarrow (\forall y)(|y| = n \Rightarrow 10 * y \notin C)$$

for each odd n.)

Positive relativizations for the two problems considered above have been considered by Book, Long and Selman [1984], [1985]. In particular, with notation as on p. 217,

$$\mathrm{NP} = \text{co-NP} \Leftrightarrow (\forall A)(\mathrm{NP}_b^A = \text{co-NP}_b^A).$$

Conflicting relativizations have also been shown to hold for a number of other open problems about the structure of NP discussed in this section. For example:

- Every infinite NP set has an infinite P subset (Bennett and Gill [1981], Homer and Maass [1983])
- There exists an NP-simple set (Homer and Maass [1983])
- If $\mathrm{P} \neq \mathrm{NP}$, then the polynomial time m-complete sets for NP are all polynomial time isomorphic (Kurtz, Mahaney and Royer [1989], and Fenner, Fortnow and Kurtz [1992]).

For other related relativizations see Kurtz [1983a], [1985], Schöning and Book [1984], Balcázar [1985], Hartmanis and Hemachandra [1987], Homer and Selman [1989], and Torenvliet and Van Emde Boas [1989], Balcázar, Díaz and Gabarró [1990].

Exercises VIII.3.33 Complete sets for NP ∩ co-NP.

a) *There is a set polynomial time m-complete for* NP ∩ *co*-NP *if and only if* NP ∩ *co*-NP *is recursively presentable by complementary pairs, i.e. there is an r.e. list of pairs of nondeterministic Turing machines accepting complementary sets in polynomial time, and the list exhausts* NP ∩ *co*-NP. (Kowalczyk [1984]) (Hint: if $\{N_{f(e)}, N_{g(e)}\}_{e \in \omega}$ is a recursive presentation of NP ∩ co-NP, let

$$\langle x, e, n\rangle \in A \;\Leftrightarrow\; N_{f(e)} \text{ accepts } x \text{ in time } |n|.$$

Then $A \in$ NP, and $A \in$ co-NP because

$$\langle x, e, n\rangle \in \overline{A} \;\Leftrightarrow\; N_{g(e)} \text{ accepts } x \text{ in time } |n|.$$

Conversely, let A be a polynomial time m-complete set for NP ∩ co-NP. Then $B \in$ NP ∩ co-NP if and only if, for some polynomial time computable function h,

$$x \in A \;\Leftrightarrow\; h(x) \in A.$$

Let $N_{f(e)}$ be the nondeterministic Turing machine that on input x first computes $\{(e)_1\}(x)$ for $p_{(e)_2}(|x|)$ steps, and if the computation converges, then it simulates a polynomial time nondeterministic Turing machine accepting A on $\{(e)_1\}(x)$ and accepts if and only if the latter does, while if the computation does not converge then it outputs 0. Also, let $N_{g(e)}$ be a nondeterministic Turing machine defined similarly, but simulating a polynomial time nondeterministic Turing machine accepting $\overline{A}$.)

b) *There is a polynomial time m-complete set for* NP ∩ *co*-NP *if and only if there is a polynomial time T-complete set for* NP∩*co*-NP. (Gurevich [1983], Hartmanis and Immerman [1985]) (Hint: suppose A is polynomial time T-complete for NP ∩ co-NP, i.e. P^A = NP ∩ co-NP. By part a), it is enough to find an r.e. list of pairs of polynomial time nondeterministic Turing machines accepting complementary sets, and exhausting P^A. Given an r.e. list of all polynomial time deterministic Turing machines with oracle A, one obtains the first list of nondeterministic Turing machines by answering positive and negative questions to the oracle A by polynomial time nondeterministic Turing machines for A and $\overline{A}$, which exist because $A \in \mathrm{NP} \cap \text{co-NP}$. The second list is obtained from the first, by interchanging accepting and rejecting states.)

c) *There is an oracle A such that* $\mathrm{NP}^A \cap$ *co*-NP^A *has a polynomial time T-complete set*. (Hint: any oracle A such that $\mathrm{P}^A = \mathrm{NP}^A$ or $\mathrm{NP}^A = \text{co-NP}^A$.) See Hartmanis and Immerman [1985] for an oracle A such that $\mathrm{NP}^A \neq \text{co-NP}^A$, and hence $\mathrm{P}^A \neq \mathrm{NP}^A$.

d) *There is an oracle B such that* $\mathrm{NP}^B \cap$ *co*-NP^B *does not have a polynomial time T-complete set*. (Sipser [1982]) (Hint: to ensure that there is no polynomial time T-complete set, by part a) we diagonalize against any r.e. list $\{N^B_{f(e)}, N^B_{g(e)}\}_{e\in\omega}$ of complementary nondeterministic Turing machines working in polynomial time $p_{h(e)}$. Let

$$x \in A \ \Leftrightarrow\ (\exists y)(|y| = |x| \ \wedge\ y \in B).$$

Then $A \in \mathrm{NP}^B$. At stage $s+1$, to diagonalize against a single list $\{N^B_{f(e)}, N^B_{g(e)}\}_{e\in\omega}$ of nondeterministic Turing machines working in polynomial time $p_{h(e)}$, choose l such that $l+1$ is greater than the length of any element on which σ_s is defined, and $2^l > p_{h(e)}(l+1)$. Let $x = 2^l$, and simulate $N^{\sigma_s}_{f(e)}$ and $N^{\sigma_s}_{g(e)}$ for $p_{h(e)}(|x|)$ steps on input x. If the outputs are not complementary (i.e. neither 1 and 0, nor 0 and 1) then we freeze both computations so that the machines are not complementary. If the outputs are complementary, freeze the computation that accepts x, and put one element of length $l+1$ into B (so that $x \in A$) if and only if $N^B_{g(e)}$ accepts x, so that the machine $N^B_{f(e)}$ does not accept A.

Notice that if the two lists are complementary then

$$x \in \overline{A} \ \Leftrightarrow\ N^B_{g(e)} \text{ accepts } x \text{ in time } p_{h(e)}(|x|),$$

so that $A \in \text{co-NP}^B$ and hence $A \in \mathrm{NP}^B \cap \text{co-NP}^B$, but it is not accepted by the first list.)

Polynomial time degrees again ⋆

The analogy between NP and r.e. sets suggests the following definition.

Definition VIII.3.34 $\boldsymbol{\mathcal{R}}^{\mathrm{P}}_{\boldsymbol{m}}$ *and* $\boldsymbol{\mathcal{R}}^{\mathrm{P}}_{\boldsymbol{T}}$ *are the structures of polynomial time m-degrees of* NP *sets.*

Notice that $\boldsymbol{\mathcal{R}}^{\mathrm{P}}_{\boldsymbol{m}}$ *is an ideal of* $\boldsymbol{\mathcal{D}}^{\mathrm{P}}_{\boldsymbol{m}}$. It is not known whether $\boldsymbol{\mathcal{R}}^{\mathrm{P}}_{\boldsymbol{T}}$ is an ideal of $\boldsymbol{\mathcal{D}}^{\mathrm{P}}_{\boldsymbol{T}}$, but Ambos-Spies [1987] has proved that the answer does not relativize.

Proposition VIII.3.35 (Ladner [1975]) $\boldsymbol{\mathcal{R}}^{\mathrm{P}}_{\boldsymbol{m}}$ *and* $\boldsymbol{\mathcal{R}}^{\mathrm{P}}_{\boldsymbol{T}}$ *contain either only one or infinitely many degrees.*

Proof. If P = NP, then there is only one polynomial time m-degree or T-degree of NP sets.

If P $\neq$ NP, then there are at least two polynomial time m-degrees or T-degrees of NP sets, since $\mathbf{0}_m^{\mathrm{P}} = \mathbf{0}_T^{\mathrm{P}} = \mathrm{P}$, and hence there are infinitely many by the Density Theorem VIII.2.38. □

The last result can be rephrased as saying that *if* NP − P *is nonempty, then it is large* from a degree-theoretical point of view. Zimand [1993] has proved that the same holds also from a topological point of view. Lutz and Mayordomo [1996] and Ambos-Spies and Bentzien [199?] have derived structural consequences from the hypothesis that NP − P is large from the measure-theoretical and categorical points of view.

A number of other results about $\boldsymbol{\mathcal{R}}_{\boldsymbol{m}}^{\mathrm{P}}$ and $\boldsymbol{\mathcal{R}}_{\boldsymbol{T}}^{\mathrm{P}}$ follow similarly from general facts about the polynomial time degrees of recursive sets, under the hypothesis P $\neq$ NP. See Ambos-Spies [1987], [1999] for surveys.

Proposition VIII.3.36 (Ladner [1975])

1. *$\boldsymbol{\mathcal{R}}_{\boldsymbol{m}}^{\mathrm{P}}$ and $\boldsymbol{\mathcal{R}}_{\boldsymbol{m}}$ are not elementarily equivalent.*
2. *$\boldsymbol{\mathcal{R}}_{\boldsymbol{T}}^{\mathrm{P}}$ and $\boldsymbol{\mathcal{R}}_{\boldsymbol{T}}$ are not elementarily equivalent.*

Proof. If P = NP, then there is only one polynomial time m-degree of NP sets, but more than one r.e. m-degree. Similarly for the T-degrees.

If P $\neq$ NP, then there is no minimal polynomial time m-degree of NP sets (VIII.2.34), but there is a minimal r.e. m-degree (X.5.11). Moreover, splitting and density can be combined for the polynomial time T-degrees of NP sets (VIII.2.39.a), but not for the T-degrees of r.e. sets (see p. 554). □

If P = NP, then $\boldsymbol{\mathcal{R}}_{\boldsymbol{m}}^{\mathrm{P}}$ and $\boldsymbol{\mathcal{R}}_{\boldsymbol{T}}^{\mathrm{P}}$ are isomorphic. Thus the hypothesis P $\neq$ NP is necessary to prove that $\boldsymbol{\mathcal{R}}_{\boldsymbol{m}}^{\mathrm{P}}$ and $\boldsymbol{\mathcal{R}}_{\boldsymbol{T}}^{\mathrm{P}}$ are not elementarily equivalent, but it is not known whether it is also sufficient. Actually, it is not even known whether P $\neq$ NP implies that $\leq_m^{\mathrm{P}}$ and $\leq_T^{\mathrm{P}}$ differ nontrivially on the NP sets (Selman [1979] has proved that this follows from the stronger hypothesis EXP $\neq$ NEXP discussed in Section 6).

VIII.4 The Polynomial Time Hierarchy ⋆

In Sections 2 and 3 we have investigated polynomial time analogues of recursive and r.e. sets. We turn now to a polynomial time analogue of the Arithmetical Hierarchy studied in Section IV.1, whose blueprint we follow.

Truth in Bounded Quantifier Arithmetic

In IV.1.1 and IV.1.2 we have introduced two different approaches to definability of truth in Arithmetic, by considering first-order languages with equality, connectives, (unbounded) quantifiers, constants $\overline{n}$ for each number n and, respectively:

- binary function symbols $+$ and $\times$
- a relation symbol φ_R for each recursive relation R,

with the intended interpretations. In IV.1.3 we have noticed that the same class of relations is defined by both approaches.

The idea for obtaining subrecursive analogues of the Arithmetical Hierarchy is to replace:

- bounded quantifiers by weakly bounded ones
- unbounded quantifiers by bounded ones
- recursive relations by polynomial time computable ones.

The first approach above produces the **Bounded Arithmetical Hierarchy**, defining the class Δ^0_0 of **rudimentary predicates** (Smullyan [1961], Bennett [1962]), and stratifying the class of relations first-order definable in the language with only plus and times by using connectives and bounded quantifiers. Some of the properties of the Bounded Arithmetical Hierarchy have been considered in IV.1.7, and we have noticed there that it is not known whether the hierarchy collapses.

We now consider the second approach.

Definition VIII.4.1 (Meyer and Stockmeyer [1972], Karp [1972]) *Let $\mathcal{L}^*_{\mathrm{P}}$ be the first-order language with equality, augmented with constants $\overline{n}$ for each number n, and a relation symbol φ_R for each polynomial time computable relation R, and let $\mathcal{A}^*_{\mathrm{P}}$ be the intended structure for $\mathcal{L}^*_{\mathrm{P}}$, i.e. the natural numbers with all the polynomial time computable relations.*

Given a closed formula φ of $\mathcal{L}^*_{\mathrm{P}}$, $\mathcal{A}^*_{\mathrm{P}} \models \varphi$ is defined inductively as usual, starting from

$$\mathcal{A}^*_{\mathrm{P}} \models \varphi_R(\overline{x}_1, \ldots, \overline{x}_n) \quad \Leftrightarrow \quad R(x_1, \ldots, x_n).$$

An n-ary relation P is in the **Polynomial Time Hierarchy** *if, for some formula φ of $\mathcal{L}^*_{\mathrm{P}}$ with n free variables and only bounded quantifiers,*

$$P(x_1, \ldots, x_n) \;\Leftrightarrow\; \mathcal{A}^*_{\mathrm{P}} \models \varphi(\overline{x}_1, \ldots, \overline{x}_n).$$

Briefly stated, the Polynomial Time Hierarchy consists of the relations definable in First-Order Arithmetic using only bounded quantifiers, with the polynomial time computable relations as parameters.

The Bounded Arithmetical Hierarchy is obviously contained in the Polynomial Time Hierarchy ($\Delta_0^0 \subseteq \mathrm{PH}$), but it is not known whether the two coincide, i.e. whether the analogue of IV.1.3 holds. By sufficiently expanding the language used for the Bounded Arithmetical Hierarchy, Buss [1986] has defined a version of it that does coincide with the Polynomial Time Hierarchy.

From closure of Δ_0^0 under bounded quantifiers (see IV.1.7.b), it follows that

$$\mathrm{P} \subseteq \Delta_0^0 \ \Rightarrow\ \mathrm{PH} = \Delta_0^0.$$

While it is not known whether $\mathrm{P} \subseteq \Delta_0^0$, Nepomniaschki [1970a] and Volger [1984] have proved that

$$\mathrm{LOGSPACE} \subseteq \Delta_0^0.$$

Moreover, Wrathall [1976] has characterized Δ_0^0 as the *Linear Time Hierarchy*, defined in the same way as PH, but with linear time computable matrices and linearly bounded quantifiers (see Hájek and Pudlák [1993] for a proof).

Types of bounded quantifiers ⋆

For the definition of the Polynomial Time Hierarchy, we used the standard bounded quantifiers of the form

$$(\forall z)_{z \le x} \quad \text{and} \quad (\exists z)_{z \le x}. \tag{VIII.1}$$

In the literature, one often uses bounded quantifiers of the form

$$(\forall z)_{|z| \le |x|} \quad \text{and} \quad (\exists z)_{|z| \le |x|}. \tag{VIII.2}$$

The two forms are equivalent modulo polynomial time computable functions, since

$$|z| \le |x| \ \Leftrightarrow\ z < 2^{|x|}$$

and

$$2^{|x|-1} \le x < 2^{|x|}.$$

Indeed, the binary representation of $x \neq 0$ consists of a 1 followed by a sequence of length $|x| - 1$ of 0's and 1's. The bounds $2^{|x|-1}$ and $2^{|x|} - 1$ correspond to the two extreme cases of a sequence consisting only of 0's or only of 1's.

The following apparently stronger forms of bounded quantifiers are also reducible to the previous forms:

$$(\forall z)_{|z| \le p(|x|)} \quad \text{and} \quad (\exists z)_{|z| \le p(|x|)}, \tag{VIII.3}$$

where p is a polynomial, and

$$(\forall z)_{z \leq f(x)} \quad \text{and} \quad (\exists z)_{z \leq f(x)}, \tag{VIII.4}$$

where f is a polynomial time computable function.

Equation VIII.3 can be reduced to VIII.2 by the same argument as in IV.1.7.b. Equation VIII.4 is reducible to VIII.3 by noticing that, on the one hand,

$$|z| \leq p(|x|) \Leftrightarrow z < 2^{p(|x|)}$$

and $2^{p(|x|)}$ is polynomial time computable (VIII.2.4); on the other hand, every polynomial time computable function $f(x)$ is bounded by $2^{p(|x|)}$, for some polynomial p (VIII.2.16).

The Polynomial Time Hierarchy

As in Chapter IV, we can classify relations in the Polynomial Time Hierarchy by putting them into normal form, and counting the number of alternating quantifiers.

Proposition VIII.4.2 *The following transformations of quantifiers are permissible (up to logical equivalence):*

1. *permutation of bounded quantifiers of the same type*
2. *contraction of bounded quantifiers of the same type.*

Proof. Part 1 is obvious. Part 2 can be accomplished by codifying the various quantified variables into a single variable as in IV.1.4, by means of a polynomial time coding and decoding mechanism (see the proof of VIII.1.18). □

Proposition VIII.4.3 Prenex Normal Form (Wrathall [1976]). *Any relation in the Polynomial Time Hierarchy is equivalent to one with a list of alternated bounded quantifiers in the prefix, and a polynomial time computable matrix.*

Proof. Similar to IV.1.5. □

Definition VIII.4.4 The Polynomial Time Hierarchy (Meyer and Stockmeyer [1972], Karp [1972], Wrathall [1976])

1. $\boldsymbol{\Sigma}_{\boldsymbol{n}}^{\mathbf{P}}$ *is the class of relations definable over* $\mathcal{A}_{\mathrm{P}}^*$ *by a formula of* $\mathcal{L}_{\mathrm{P}}^*$ *in prenex form with a polynomial time computable matrix and n bounded quantifier alternations in the prefix, the outer bounded quantifier being existential.*

2. $\mathbf{\Pi}^{\mathbf{P}}_{\boldsymbol{n}}$ *is defined similarly, with the outer bounded quantifier being universal.*

3. $\mathbf{\Delta}^{\mathbf{P}}_{\boldsymbol{n}}$ *is* $\Sigma^{\mathrm{P}}_n \cap \Pi^{\mathrm{P}}_n$*, i.e. the class of relations definable in both the n-quantifier forms.*[3]

4. **PH** *is the class of relations in the Polynomial Time Hierarchy.*

By extension, we will call a *formula* Σ^{P}_n or Π^{P}_n, if it is in prenex normal form, with n bounded quantifier alternations in the prefix, the outer one being, respectively, existential or universal.

Note also that, by contraction of bounded quantifiers, n bounded quantifier alternations are equivalent to n alternated bounded quantifiers.

Second-Order Logic on finite domains ⋆

The Polynomial Hierarchy has just been defined by considering a first-order arithmetical language with bounded number quantifiers. We now discuss an alternative definition obtained by considering a second-order logical language with relations restricted to finite domains.

Definition VIII.4.5 (Fagin [1974]) *Let $\mathcal{L}^2_{\mathrm{P}}$ be the second-order language with first-order equality, augmented with constants $\overline{n}$ for each number n, and let $\mathcal{A}^2_{\mathrm{P}}$ be the intended structure for $\mathcal{L}^2_{\mathrm{P}}$, i.e. the natural numbers with all their finite relations.*

Given a closed formula φ of $\mathcal{L}^2_{\mathrm{P}}$, $\mathcal{A}^2_{\mathrm{P}} \models \varphi$ is defined inductively as usual, starting from

$$\mathcal{A}^2_{\mathrm{P}} \models \overline{x} = \overline{y} \;\Leftrightarrow\; x = y.$$

An n-ary relation P is **second-order definable on finite domains** *if, for some formula φ of $\mathcal{L}^2_{\mathrm{P}}$ with n free first-order variables and second-order quantifiers restricted to relations on finite domains,*

$$P(x_1, \ldots, x_n) \;\Leftrightarrow\; \mathcal{A}^2_{\mathrm{P}} \models \varphi(\overline{x}_1, \ldots, \overline{x}_n).$$

As in Chapter IV, we can classify second-order definable relations on finite domains by putting them into prenex normal form, with a prefix of second-order quantifiers and a first-order matrix, and counting the number of alternations of second-order quantifiers. The following result, stated without proof, shows that this classification provides an alternative characterization of the Polynomial Time Hierarchy.

[3]In the literature, Δ^{P}_n is sometimes defined as the class of relations polynomial time Turing reducible to a Σ^{P}_n or a Π^{P}_n relation (in particular, $\Delta^{\mathrm{P}}_1 = \mathrm{P}$). In this case one defines Δ^{P}_n not as a direct analogue of Δ^0_n, but as an indirect analogue via Part 1 of Post's Theorem IV.1.14 (see VIII.4.13 for an analogue of Part 2).

Theorem VIII.4.6 Logical Characterization of PH (Fagin [1974])

1. *For any $n \geq 1$, a relation is in Σ_n^{P} if and only if it is second-order definable on finite domains by a formula in prenex normal form with first-order matrix and n second-order quantifier alternations in the prefix, the outer second-order quantifier being existential.*

2. *Similarly for Π_n^{P} relations, with the outer second-order quantifier being universal.*

3. *A relation is in* PH *if and only if it is second-order definable on finite domains.*

In particular, NP is an analogue not only of Σ_1^0 in arithmetic, but also of Σ_1^1 in logic. In the first case the unbounded range of the first-order existential quantifier is replaced by a bounded finite range. In the second case the unrestricted second-order existential quantifier is replaced by a quantifier restricted to relations on finite domains.

The result just quoted suggests the possibility of obtaining similar logical characterizations of other complexity classes. The first obvious thought would be to consider the first-order definable relations, but it turns out that they form only a small proper subset of LOGSPACE (Ajtai [1983], Immerman [1983]).

To characterize classes smaller than NP, there are two possible ways: either to strengthen first-order logic by enlarging its language, or to weaken existential second-order logic by restricting the first-order language used in the matrix. By following the first way, it can be proved that NLOGSPACE and P are captured by adding, respectively, a transitive closure operator (Immerman [1983]) or a least-fixed point operator (Immerman [1982], Vardi [1982]) to the first-order language with equality and order. By following the second way, it can be proved that NLOGSPACE and P are captured by restricting the matrices of existential second-order formulas to, respectively, Krom or Horn formulas in the first-order language with equality, order and successor (Grädel [1991]).

To characterize classes larger than PH, such as those considered in the next sections, there are again two possible ways: to strengthen second-order logic either by enlarging its language, or by stepping to higher-order logics.

For introductory surveys on this subject, see Immerman [1989], Leivant [1989] and Fagin [1993]. For a treatment, see Immerman [1999].

The levels of the Polynomial Time Hierarchy

The first levels of the Polynomial Time Hierarchy are inhabited by old friends. First, the level 0 obviously consists of the polynomial time computable relations (because no bounded quantifier is involved), i.e.

$$\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \Delta_0^{\mathrm{P}} = \mathrm{P}.$$

More interestingly,

$$\begin{array}{rcl} \Delta_1^{\mathrm{P}} & = & \mathrm{NP} \cap \text{co-NP} \\ \Sigma_1^{\mathrm{P}} & = & \mathrm{NP}. \end{array}$$

The latter follows from VIII.3.3 and the discussion on bounded quantifiers on p. 225, and it implies the former.

Proposition VIII.4.7 Closure Properties (Wrathall [1976])

1. *R is Σ_n^{P} if and only if $\neg R$ is Π_n^{P}*
   R is Π_n^{P} if and only if $\neg R$ is Σ_n^{P}
2. *Δ_n^{P} is closed under negations*
3. *Σ_n^{P}, Π_n^{P} and Δ_n^{P} are closed under conjunction, disjunction and weak bounded quantification*
4. *for $n \geq 1$, Σ_n^{P} is closed under bounded existential quantification, and Π_n^{P} is closed under bounded universal quantification*
5. *the bounded universal quantification of a Σ_n^{P} relation is Π_{n+1}^{P}, and the bounded existential quantification of a Π_n^{P} relation is $\Sigma_{n+1}^{\mathrm{P}}$.*

Proof. As in IV.1.8, by using the fact that polynomial time computable relations are closed under weak bounded quantification (VIII.2.6). $\Box$

Theorem VIII.4.8 Enumeration Theorem (Meyer and Stockmeyer [1972], Karp [1972], Wrathall [1976]) *For each $n, m \geq 1$ there is an $m+1$-ary Σ_n^{P} relation that enumerates the m-ary Σ_n^{P} relations. Similarly for Π_n^{P}.*

Proof. This is simply a consequence of the Normal Form Theorem for NP relations VIII.3.3, according to which there is an $m+1$-ary Σ_1^{P} relation enumerating the m-ary Σ_1^{P} relations. Then its negation enumerates the m-ary Π_1^{P} relations.

Inductively, let $R(e, x, \vec{z})$ be an $m+2$-ary Σ_n^{P} relation enumerating the $m+1$-ary Σ_n^{P} relations. Then $\forall x R(e, x, \vec{z})$ is an $m+1$-ary Π_{n+1}^{P} relation enumerating the m-ary Π_{n+1}^{P} relations, and its negation enumerates the m-ary $\Sigma_{n+1}^{\mathrm{P}}$ relations. $\Box$

Definition VIII.4.9 *A set A is* $\boldsymbol{\Sigma_n^{\mathrm{P}}}$**-complete** *if it is Σ_n^{P}, and every Σ_n^{P} set is polynomial time m-reducible to it.* $\boldsymbol{\Pi_n^{\mathrm{P}}}$**-complete** *sets are defined similarly.*

The concept of Σ_n^{P}-completeness is the analogue, at level n, of the concept of polynomial time m-completeness for NP sets. By induction we thus have:

Proposition VIII.4.10 *For each $n \geq 1$, Σ_n^{P}-complete and Π_n^{P}-complete sets exist.*

Proof. Let $\emptyset_{\mathrm{P}} = \emptyset$, and define $\emptyset_{\mathrm{P}}^{(n+1)}$ inductively as the relativization of $\mathcal{K}_0^{\mathrm{NP}}$ to $\emptyset_{\mathrm{P}}^{(n)}$. Then $\emptyset_{\mathrm{P}}^{(n)}$ is Σ_n^{P}-complete. $\square$

The next result we proved for the Arithmetical Hierarchy was the Hierarchy Theorem IV.1.13, showing that the hierarchy does not collapse. In particular, for each $n \geq 1$,

$$\Delta_n^0 \subset \Sigma_n^0 \cup \Pi_n^0 \subset \Delta_{n+1}^0.$$

The analogue for the Polynomial Time Hierarchy is not known. More precisely, it is not known whether, for any $n \geq 1$, any of the inclusions

$$\Delta_n^{\mathrm{P}} \subseteq \Sigma_n^{\mathrm{P}} \cup \Pi_n^{\mathrm{P}} \subseteq \Delta_{n+1}^{\mathrm{P}}$$

is proper.

Theorem VIII.4.11 First Collapse Theorem for PH (Stockmeyer [1976]) *If $\Sigma_n^{\mathrm{P}} = \Pi_n^{\mathrm{P}}$ or $\Sigma_n^{\mathrm{P}} = \Sigma_{n+1}^{\mathrm{P}}$, then* $\mathrm{PH} = \Delta_n^{\mathrm{P}}$. *In particular:*

1. *if* $\mathrm{P} = \mathrm{NP}$*, then* $\mathrm{PH} = \Delta_0^{\mathrm{P}}$
2. *if* $\mathrm{NP} = co\text{-}\mathrm{NP}$*, then* $\mathrm{PH} = \Delta_1^{\mathrm{P}}$.

Proof. First, notice that if $\Sigma_n^{\mathrm{P}} = \Sigma_{n+1}^{\mathrm{P}}$, then $\Pi_n^{\mathrm{P}} \subseteq \Sigma_{n+1}^{\mathrm{P}} = \Sigma_n^{\mathrm{P}}$ and, symmetrically, $\Sigma_n^{\mathrm{P}} \subseteq \Pi_n^{\mathrm{P}}$, i.e. $\Sigma_n^{\mathrm{P}} = \Pi_n^{\mathrm{P}}$.

It is thus enough to prove by induction that if $\Sigma_n^{\mathrm{P}} = \Sigma_{n+1}^{\mathrm{P}}$, then $\Sigma_n^{\mathrm{P}} = \Sigma_{n+m}^{\mathrm{P}}$ for every m. Then $\mathrm{PH} = \Sigma_n^{\mathrm{P}} = \Pi_n^{\mathrm{P}} = \Delta_n^{\mathrm{P}}$.

- For $m = 0$, there is nothing to prove.
- For $n + 1$, suppose $\Sigma_n^{\mathrm{P}} = \Sigma_{n+m}^{\mathrm{P}}$. By taking negations, $\Pi_n^{\mathrm{P}} = \Pi_{n+m}^{\mathrm{P}}$. By closing under bounded existential quantification, $\Sigma_{n+1}^{\mathrm{P}} = \Sigma_{n+m+1}^{\mathrm{P}}$. But $\Sigma_n^{\mathrm{P}} = \Sigma_{n+1}^{\mathrm{P}}$, since the latter is the bounded existential quantification of $\Pi_n^{\mathrm{P}} = \Sigma_n^{\mathrm{P}}$, and the former is closed under such quantifications. Thus $\Sigma_n^{\mathrm{P}} = \Sigma_{n+m+1}^{\mathrm{P}}$. $\square$

A similar argument shows that a log-space version of the Polynomial Time Hierarchy such that (with obvious notation) $\Sigma_1^{\mathrm{LOGSPACE}} = \mathrm{NLOGSPACE}$ would collapse at level 1, because of VIII.1.14. Such a hierarchy was indeed defined by Chandra, Kozen and Stockmeyer [1981], and shown to collapse (before VIII.1.14

was available, and in a weaker manner) by Toda [1987], Schöning and Wagner [1988], and Jenner, Kirsig and Lange [1989].

A polynomial time analogue BPH $\subseteq \Delta_2^{\mathrm{P}}$ of the Boolean Hierarchy BH $\subseteq \Delta_2^0$ (see IV.1.18) has been studied by Cai, Gundermann, Hartmanis, Hemachandra, Sewelson, Wagner and Wechsung [1988], [1989]. Kadin [1988] has extended VIII.4.11 by showing that *if* BPH *collapses, then* PH $= \Delta_3^{\mathrm{P}}$.

Exercise VIII.4.12 *If* NP $\neq$ *co*-NP *then* $\Sigma_1^{\mathrm{P}} \cup \Pi_1^{\mathrm{P}} \subset \Delta_2^{\mathrm{P}}$. (Long) (Hint: let $A \in$ NP − co-NP. Then $A \oplus \overline{A} \in \Delta_2^{\mathrm{P}} - (\mathrm{NP} \cup \text{co-NP})$.)

As for the Arithmetical Hierarchy, we have defined the Polynomial Time Hierarchy in a semantical way, by iterating bounded quantifiers over polynomial time computable matrices. The next result is the analogue of Part 2 of Post's Theorem IV.1.14 and gives a different, purely complexity-theoretical, definition of the Polynomial Time Hierarchy.

Proposition VIII.4.13 (Meyer and Stockmeyer [1972], Karp [1972], Wrathall [1976]) *A relation is* $\Sigma_{n+1}^{\mathrm{P}}$ *if and only if it is* NP *in a* Σ_n^{P} *or a* Π_n^{P} *relation.*

Proof. As in part 2 of IV.1.14, using a relativized version of $\Sigma_1^{\mathrm{P}} = \mathrm{NP}$, and the fact that if D_u and D_v are the sets of queries asked positively and negatively to the oracle A during a (nondeterministic) polynomial time computation, then u and v are bounded by a polynomial p in the length of the input x, and the expressions $D_u \subseteq A$ and $D_v \subseteq \overline{A}$ are Π_{n+1}^{P} if A is Σ_n^{P}. □

Obviously, the analogue of part 1 of Post's Theorem (namely that a relation is $\Delta_{n+1}^{\mathrm{P}}$ if and only if it is polynomial time Turing reducible to a Σ_n^{P} or a Π_n^{P} relation) is an open problem, since it would imply in particular $\Delta_1^{\mathrm{P}} = \mathrm{P}$, i.e. P = NP ∩ co-NP.[4]

Relativizations

From VIII.4.11 we already know that there are oracles A such that PH^A collapses (for example, if $\mathrm{P}^A = \mathrm{NP}^A$ as in VIII.3.28, then PH^A collapses to $\Delta_0^{\mathrm{P},A}$; and if $\mathrm{NP}^A = \text{co-NP}^A$ as in VIII.3.31, then PH^A collapses to $\Delta_1^{\mathrm{P},A}$). This result has been refined and complemented, as follows:

[4]If one defines $\Delta_{n+1}^{\mathrm{P}}$ as in Footnote 3 on p. 227, one only has

$$\Sigma_n^{\mathrm{P}} \cup \Pi_n^{\mathrm{P}} \subseteq \Delta_{n+1}^{\mathrm{P}} \subseteq \Sigma_{n+1}^{\mathrm{P}} \cap \Pi_{n+1}^{\mathrm{P}},$$

and the open problem above then translates into whether the rightmost inclusion is an equality. Of course, the same problem occurs for the leftmost inclusion.

- *There are oracles A_n such that* PH^{A_n} *collapses to level n but not before* (Baker, Gill and Solovay [1975], Baker and Selman [1979], Heller [1984], Ko [1988]).

- *There is an oracle B such that* PH^B *does not collapse* (Furst, Saxe and Sipser [1984], Yao [1985], Håstad [1986], [1987], Cai [1986], Babai [1987]).

Since all known proofs of these results use methods outside the scope of this book, we cannot prove them here.

In a different direction, the proofs of VIII.3.28 and VIII.3.29 showed that *there are sparse oracles separating* P *and* NP*, and sparse oracles collapsing them.* Thus, even by restricting ourselves to sparse oracles, we still have conflicting relativizations of the problem of whether P = NP. The next result shows that this is not the case for the problem of whether PH collapses.

Proposition VIII.4.14 Positive Relativization of the PH Collapse (Long and Selman [1986])

$$\mathrm{PH}\ \textit{collapses} \Leftrightarrow (\forall \textit{sparse}\, A)(\mathrm{PH}^A\ \textit{collapses}).$$

Proof. One direction is trivial, since $\emptyset$ is sparse. For the converse, we show that if A is sparse, $n \geq 2$ and $\Sigma^{\mathrm{P}}_{n+1} = \Sigma^{\mathrm{P}}_n$, then $\Sigma^{\mathrm{P},A}_{n+1} = \Sigma^{\mathrm{P},A}_n$. By the relativized Collapse Theorem VIII.4.11, this proves that if PH collapses, then so does PH^A.

Let $B \in \Sigma^{\mathrm{P},A}_{n+1}$, i.e.

$$x \in B \Leftrightarrow (\exists x_1)_{|x_1| \leq p_1(|x|)} (\forall x_2)_{|x_2| \leq p_2(|x|)} \cdots R^A(x, x_1, \ldots, x_{n+1}),$$

where the p_i's are polynomials, and R^A is computable in polynomial time p_0 with oracle A. The idea is to replace the oracle A by a finite oracle σ, thus turning the previous $\Sigma^{\mathrm{P},A}_{n+1}$ oracle expression into a $\Sigma^{\mathrm{P}}_{n+1}$, and hence Σ^{P}_n, absolute one. This will be done at the cost of an additional $\Sigma^{\mathrm{P},A}_2$ clause, which is harmless because of the hypothesis $n \geq 2$.

The time needed to compute the predicate $R^A(x, x_1, \ldots, x_{n+1})$ is bounded by $p_0(|x|, |x_1|, \ldots, |x_{n+1}|)$, and hence so is the length of any query to the oracle A. But $|x_i| \leq p_i(|x|)$, and so the length of any query to the oracle A is bounded by the polynomial

$$p(|x|) = p_0(|x|, p_1(|x|), \ldots, p_{n+1}(|x|)).$$

Since A is sparse, there is a polynomial q such that

$$|A \cap \{0, \ldots, 2^n\}| \leq q(n).$$

Thus A has at most $q(p(|x|))$ elements of length bounded by $p(|x|)$. These elements can then be coded by a string σ of polynomial length $r(|x|)$, for some polynomial r. Then

$$\begin{aligned} x \in B \ \Leftrightarrow \ & (\exists \sigma)_{|\sigma| \leq r(|x|)}[\sigma \text{ codes } A \cap \{0, \ldots, 2^{p(|x|)}\} \wedge \\ & (\exists x_1)_{|x_1| \leq p_1(|x|)} (\forall x_2)_{|x_2| \leq p_2(|x|)} \cdots R^{\sigma}(x, x_1, \ldots, x_{n+1})]. \end{aligned}$$

The clause 'σ codes $A \cap \{0, \ldots, 2^{p(|x|)}\}$' can be expressed with only one quantifier, being equivalent to

$$(\forall y)_{|y| \leq p(|x|)}(y \in \sigma \ \Leftrightarrow \ y \in A).$$

It is thus $\Pi_1^{\mathrm{P},A}$ and hence, since $n \geq 2$, also $\Sigma_n^{\mathrm{P},A}$.

The second clause of the expression of B above is $\Sigma_{n+1}^{\mathrm{P}}$, because the oracle A has been replaced by a finite string. By the hypothesis, it is thus Σ_n^{P} and hence also $\Sigma_n^{\mathrm{P},A}$.

The complexity of the expression in square brackets is thus $\Sigma_n^{\mathrm{P},A}$, and is not increased by the outer bounded quantifier $\exists \sigma$. Then $B \in \Sigma_n^{\mathrm{P},A}$. □

Balcázar, Book and Schöning [1986] have strengthened the trivial direction of the last result, by showing that *if* PH^A *collapses for some sparse A, then so does* PH. It follows that

$$\text{PH } \textit{does not collapse} \ \Leftrightarrow \ (\forall \textit{sparse } A)(\text{PH}^A \ \textit{does not collapse}).$$

The next exercises explore different uses of relativization.

Exercises VIII.4.15 The Polynomial Time Jump Hierarchy. (Schöning [1983], Ko and Schöning [1985]) The following exercises provide a classification of NP analogous to that of the r.e. sets in terms of the degree of their jumps (see XI.1.9). Notice that, for every $A \in \text{NP}$,

$$\Sigma_n^{\mathrm{P}} \subseteq \Sigma_n^{\mathrm{P},A} \subseteq \Sigma_{n+1}^{\mathrm{P}}.$$

The next definition isolates the sets with extremal behavior.

$$\begin{array}{lcl} \mathbf{L}_n^{\mathrm{P}} & = & \{A \in \text{NP} : \Sigma_n^{\mathrm{P},A} \subseteq \Sigma_n^{\mathrm{P}}\} \\ \mathbf{H}_n^{\mathrm{P}} & = & \{A \in \text{NP} : \Sigma_{n+1}^{\mathrm{P}} \subseteq \Sigma_n^{\mathrm{P},A}\} \\ \mathbf{I}^{\mathrm{P}} & = & \{A \in \text{NP} : A \notin \bigcup_{n \in \omega}(\mathbf{L}_n^{\mathrm{P}} \cup \mathbf{H}_n^{\mathrm{P}})\}. \end{array}$$

Sets in $\mathbf{L}_n^{\mathrm{P}}$, $\mathbf{H}_n^{\mathrm{P}}$ and I^{P} are called, respectively, $\mathbf{low}_n^{\mathrm{P}}$, $\mathbf{high}_n^{\mathrm{P}}$ and $\mathbf{intermediate}^{\mathrm{P}}$.

a) $\mathbf{L}_1^{\mathrm{P}} = \Delta_1^{\mathrm{P}}$. (Hint: if $A \in \mathbf{L}_1^{\mathrm{P}}$, then $A \in \text{NP}$ by definition. Also, $\overline{A}$ is obviously in $\Sigma_1^{\mathrm{P},A}$, since $x \in \overline{A}$ can be answered by a direct query to the oracle A, and then by lowness $\overline{A}$ is in Σ_1^{P}, i.e. A is co-NP.

Conversely, let $A \in \text{NP} \cap \text{co-NP}$. To show that A is in $\mathbf{L}_1^{\text{P}}$, one has to show that the polynomial time m-complete set for NP relative to A is in NP. To find out whether N_e^A accepts x in time $|n|$, one simulates N_e^A on x for $|n|$ steps, and whenever the oracle A is queried for a number z, one simultaneously runs two nondeterministic Turing machines running in polynomial time, one for A and one for $\overline{A}$, on z. Since at most n queries are made, the whole procedure takes only polynomial time.)

b) *A sparse* NP *set is in* $\mathbf{L}_2^{\text{P}}$. (Hint: the proof of VIII.4.14 reduces the problem to showing that the $\Pi_1^{\text{P},A}$ clause $(\forall y)_{|y| \leq p(|x|)}(y \in \sigma \Leftrightarrow y \in A)$ is Σ_2^{P}. Using the NP form of A is sufficient for the implication $y \in A \Rightarrow y \in \sigma$, because it gives a Π_1^{P} expression for it. For the opposite implication, where the NP form of A would give only a Π_2^{P} expression, one uses the fact that $y \in A \Leftrightarrow (\exists z)_{|z| \leq s(|y|)} Q(y,z)$ for some polynomial s and polynomial time computable Q. Since $|y|$ was bounded by a polynomial in $|x|$, so is $|z|$. Then one can say that every element of σ is in A by saying that there is a string τ coding the same number of elements as σ and such that, for any i, if y and z are the i-th elements of σ and τ, then $Q(y,z)$. This provides the required Σ_2^{P} form.)

c) *If* $\Sigma_n^{\text{P}} = \Sigma_{n+1}^{\text{P}}$, *then* $\text{NP} = \mathbf{L}_n^{\text{P}} = \mathbf{H}_n^{\text{P}}$. (Hint: by the inclusions noticed at the beginning.)

d) *If* $\Sigma_n^{\text{P}} \neq \Sigma_{n+1}^{\text{P}}$, *then* $\mathbf{L}_n^{\text{P}} \cap \mathbf{H}_n^{\text{P}} = \emptyset$. (Hint: by contrapositive.)

e) PH *does not collapse if and only if* $(\bigcup_{n \in \omega} \mathbf{L}_n^{\text{P}}) \cap (\bigcup_{n \in \omega} \mathbf{H}_n^{\text{P}}) = \emptyset$. (Hint: by parts c) and d).)

For additional results on the Polynomial Time Jump Hierarchy see Schöning [1986], Ko [1991], Allender and Hemachandra [1992]. For a polynomial time analogue of the Generalized Jump Hierarchy (XI.1.24), see Balcázar, Book and Schöning [1986a].

Exercise VIII.4.16 Sparse oracles. VIII.4.14 suggests consideration of the **Advice Polynomial Time Hierarchy**, defined by Yap [1983] as

$$\text{PH/poly} = \bigcup_{A \text{ sparse}} \text{PH}^A.$$

The classes Σ_n^{P}/poly and Π_n^{P}/poly are defined similarly. Sets in

$$P/\text{poly} = \bigcup_{A \text{ sparse}} \text{P}^A$$

are said to have **polynomial size circuits** (because of their original definition in terms of circuits).

a) *An* NP *set with polynomial size circuits is in* $\mathbf{L}_3^{\text{P}}$. (Ko and Schöning [1985]) (Hint: let A be sparse and let $C \in NP$ be computed by a deterministic Turing machine M^A working in polynomial time. Then any $\Sigma_3^{\text{P},C}$ expression is also $\Sigma_3^{\text{P},A}$, and as in VIII.4.15.b, it can be reduced to Σ_3^{P} form by replacing the sparse oracle A by a finite string σ of polynomial length. However, we cannot eliminate the reference to the oracle A directly, for example by saying that σ codes the elements of A up to a given length, because A is not known. But A is used only to answer queries of a bounded length about C, and it is thus enough to say that σ correctly replaces A as far as those elements are concerned. Since $C \in \mathrm{NP}$, the expression 'for every element y up to the given length, $y \in C$ if and only if M^σ says so' is Π_2^{P}, and the existential quantifier on σ brings the total complexity to Σ_3^{P}.)

b) *If there is a sparse polynomial time m-complete set for* NP*, then* $\mathrm{PH} = \Delta_0^{\mathrm{P}}$. (Mahaney [1982]) (Hint: see VIII.3.14.)

c) *If there is a sparse polynomial time T-complete set for* NP*, then* $\mathrm{PH} = \Delta_2^{\mathrm{P}}$. (Karp and Lipton [1980], Sipser) (Hint: if $\mathrm{NP} \subseteq \mathrm{P}^A$ and $A \in \mathrm{NP}$, then $A \in \mathbf{H}_0^{\mathrm{P}}$ by definition. If A is sparse, then $A \in \mathbf{L}_2^{\mathrm{P}}$ by VIII.4.15.b. The result follows from VIII.4.15.d and VIII.4.11.)

d) *If there is a polynomial time T-complete set for* NP *with polynomial size circuits, then* $\mathrm{PH} = \Delta_3^{\mathrm{P}}$. (Karp and Lipton [1980], Sipser) (Hint: as in part b), by part a).)

By a different proof, Karp and Lipton [1980] have improved the collapse in part d) to Δ_2^{P}. For additional results, see Mahaney [1982], [1986] and Kadin [1987].

Exercises VIII.4.17 Weak polynomial reducibilities. By analogy with IV.1.20, we define

$$A \leq_{\Delta_n^{\mathrm{P}}} B \Leftrightarrow A \in \Delta_n^{\mathrm{P},B}.$$

Thus $\leq_{\Delta_0^{\mathrm{P}}}$ coincides with $\leq_T^{\mathrm{P}}$ (since $\Delta_0^{\mathrm{P}} = \mathrm{P}$).

a) $\leq_{\Delta_1^{\mathrm{P}}}$ *is transitive.* (Long [1982a]) (Hint: as in VIII.4.13.) Thus $\leq_{\Delta_1^{\mathrm{P}}}$ induces a reducibility, called **strong nondeterministic Turing reducibility** and indicated by $\leq^{SN}$ in the literature.

b) $\mathbf{H}_1^{\mathrm{P}}$ *contains exactly the SN-complete sets for* NP. (Long [1982a]) (Hint: if $\mathcal{K}_0^{\mathrm{NP}} \leq^{SN} A$, then $\mathcal{K}_0^{\mathrm{NP}} \in \Delta_1^{\mathrm{P},A}$. In particular, $\Pi_1^{\mathrm{P}} \subseteq \Sigma_1^{\mathrm{P},A}$ and $\Sigma_2^{\mathrm{P}} \subseteq \Sigma_1^{\mathrm{P},A}$, i.e. $A \in \mathbf{H}_1^{\mathrm{P}}$, and conversely.)

c) *If $n > 1$ and $\Sigma_n^{\mathrm{P}} \neq \Pi_n^{\mathrm{P}}$, then $\leq_{\Delta_n^{\mathrm{P}}}$ is not transitive.* (Hint: as in IV.1.20.)

VIII.5 Polynomial Space ⋆

Space bounds have already been considered in Sections VIII.1 and VIII.2. More precisely, we have analyzed the classes of sets computable in (deterministic or nondeterministic) constant space (VIII.1.5), log-space (VIII.1.13) and $\log^n$-space (VIII.2.23).

After the intermission of Sections VIII.3 and VIII.4, that allowed for a consideration of polynomial time bounds and of the Polynomial Time Hierarchy, we now return to space bounds and resume from where we stopped.

Linear space ⋆

Since the exponential function $2^{|x|}$ grows faster than any power $|x|^n$, the logarithmic function $\log|x|$ grows slower than any root $\sqrt[n]{|x|}$, and thus any power

$(\log|x|)^n$ of the logarithmic function grows slower than the identity function $|x|$. We thus start with the consideration of the space bound $|x|$ or, equivalently (by the Linear Speed-Up Theorem for space VII.4.8), of any linear space bound $a \cdot |x| + b$.

Turing machines working in linear space have been considered by Myhill [1960], and are called **linearly bounded automata**. If we let

$$\mathrm{LINSPACE} = \mathrm{SPACE}[\,|x|\,],$$

then

$$\mathrm{POLYLOGSPACE} \subset \mathrm{LINSPACE}$$

(the inclusion follows from the previous discussion, and it is strict by the Space Hierarchy Theorem VII.4.13). A syntactic characterization of LINSPACE has been given by Ritchie [1963] (as $(\mathcal{E}_2)_*$, see p. 306).

If we let

$$\mathrm{NLINSPACE} = \mathrm{NSPACE}[\,|x|\,],$$

then obviously

$$\mathrm{LINSPACE} \subseteq \mathrm{NLINSPACE},$$

but it is not known whether equality holds.[5] Savitch [1970] has shown, by a proof as in VIII.6.6, that

$$\mathrm{LOGSPACE} = \mathrm{NLOGSPACE} \Rightarrow \mathrm{LINSPACE} = \mathrm{NLINSPACE}.$$

The significance of the LINSPACE = NLINSPACE problem is discussed in Hartmanis and Hunt [1974].

LINSPACE and NLINSPACE are both closed under complement, the former as a deterministic class and the latter by VII.4.23.

Exercise VIII.5.1 Context-sensitive grammars. Recall the definition of grammar from VIII.1.10. A **context-sensitive grammar** is a grammar whose productions are of the form

$$\sigma_1 x \sigma_2 \to \sigma_1 \tau \sigma_2,$$

where σ_1, σ_2 and τ are strings (of which τ is nonempty), and x is a variable for individual symbols (Chomsky [1956]). Thus context-sensitive grammars permit a replacement in the (possibly empty) context $\sigma_1 \,\square\, \sigma_2$.

NLINSPACE *is the class of sets generated by context-sensitive grammars*. (Landweber [1963], Kuroda [1964]). (Hint: the derivation of a string from an axiom can be simulated by a nondeterministic Turing machine that has the string on its input tape, the axiom on its working tape, and successively modifies the string on the working tape by (nondeterministically) applying productions to it, until it reaches a string equal to the input. Since context-sensitive productions never decrease the length of

[5]In the literature, LINSPACE and NLINSPACE are sometimes called LBA and NLBA.

a string, the machine can restrict its working tape to a length equal to that of the input, and it is thus a linearly bounded nondeterministic automaton.

Conversely, a set of strings accepted by a nondeterministic linearly bounded automaton can be generated by a context-sensitive grammar, whose productions simulate the instructions of the automaton. A first problem is that, since the input of the automaton must become the output of the derivation if accepted, one actually needs three kinds of productions: one turning each symbol of the input into a pair of equal symbols, thus associating with the input a copy on which one can work; one simulating the instructions of the automaton on the copy; and one restoring the input when an accepting computation is finished, by turning a pair into its first component. A second problem is that to simulate the instructions of the automaton one needs to know the current state, and this cannot be inserted among the symbols as in the usual description of instantaneous configurations (for example, as in VII.4.5), since at the end they would have to be erased, which cannot be done by context-sensitive productions: then one actually works with triples as opposed to pairs, adding states as possible third components. The final productions are context-sensitive because they always work on strings of the same length as the input.)

For more information on context-sensitive grammars see Salomaa [1973], Ginsburg [1975], Hopcroft and Ullman [1979], Lewis and Papadimitriou [1981].

The class PSPACE

Together with P and NP, the following is one of the best known complexity classes.

Definition VIII.5.2 **PSPACE** *is the class of sets computable in deterministic polynomial space in the length of the input.*

NPSPACE *is the class of sets acceptable in deterministic polynomial space in the length of the input.*

We have referred to the these two classes as one because they coincide.

Proposition VIII.5.3 (Savitch [1970], Tseitin) PSPACE = NPSPACE.

Proof. Savitch's Theorem provides a quadratic trade-off between deterministic and nondeterministic space bounds (above $\log |x|$), and $(|x|^n)^2 = |x|^{2n}$. □

Exercises VIII.5.4 Polynomial space computable functions. PSPACEF is the class of functions computable in polynomial space in the length of the input.

a) PF ⊆ PSPACEF.

b) P = PSPACE *if and only if* PF = PSPACEF. (Valiant [1976]) (Hint: one direction is trivial. For the other, let $f \in$ PSPACEF, and define

$$\langle x, y\rangle \in A \Leftrightarrow (\exists z)[y * z = f(x)].$$

Thus $\langle x, y\rangle \in A$ if y is a beginning of $f(x)$. $A \in$ PSPACE, since it is enough to compute $f(x)$ in polynomial space, and then to check whether y is a beginning of $f(x)$. Then $A \in$ P because P = PSPACE. In polynomial time we can find successive beginnings of $f(x)$, as follows. Ask if 0 is a beginning of $f(x)$; if so, $f(x) = 0$; otherwise, let $y_0 = 1$. Having y_n, ask whether $y_n * 0$ is a beginning of $f(x)$, and if so, let $y_{n+1} = y_n * 0$; otherwise, ask whether $y_{n+1} * 1$ is a beginning of $f(x)$, and if so, let $y_{n+1} = y_n * 1$; otherwise, stop. The procedure stops in polynomially many steps, because f is computable in polynomial space, and so $|f(x)|$ is bounded by a polynomial.)

Similar results also hold for P = NP and NP = PSPACE, although (for reasons discussed at the beginning of the subsection on p. 86) we prefer not to deal with nondeterministic functions. These results explain why, for the basic questions about equality of complexity classes, we restrict our attention to classes of sets (as opposed to classes of functions).

Complementing similar machine-theoretical characterizations of CONSPACE, LOGSPACE and P, Hopcroft and Ullman [1967] and Ibarra [1971] have proved that PSPACE is the class of sets computable by (non)deterministic two-way *multihead nonerasing stack automata*.

PSPACE is defined in terms of bounded *space* computations. Chandra, Kozen and Stockmeyer [1981] have characterized it in terms of bounded *time* computations, by showing that PSPACE = AP, i.e. PSPACE is the class of sets computable in polynomial time in the length of the input by alternating Turing machines.

If one defines the classes

$$\text{PSPACE}_n = \text{SPACE}\,[\,|x^n|\,]$$

(in particular, PSPACE_1 = LINSPACE), and

$$\text{NPSPACE}_n = \text{NSPACE}\,[\,|x|^n\,]$$

(in particular, NPSPACE_1 = NLINSPACE), then one immediately has a double hierarchy theorem.

Theorem VIII.5.5 Hierarchy Theorem for PSPACE.

1. $\text{PSPACE}_n \subset \text{PSPACE}_{n+1}$, *for each* n
2. $\text{NPSPACE}_n \subset \text{NPSPACE}_{n+1}$, *for each* n
3. $\text{PSPACE} = \bigcup_{n\in\omega} \text{PSPACE}_n = \bigcup_{n\in\omega} \text{NPSPACE}_n$.

Proof. Parts 1 and 2 follow from the Space Hierarchy Theorems VII.4.13 and VII.4.24, respectively.

The first equality of Part 3 is simply a restatement of the definition of PSPACE. The second equality follows from the first and VIII.5.3, since it is a restatement of the definition of NPSPACE. □

It is not known whether $\text{PSPACE}_n = \text{NPSPACE}_n$ for some n (for $n = 1$ this reduces to the problem of whether LINSPACE = NLINSPACE). Notice that this might fail for every $n \geq 1$, despite the fact that

$$\bigcup_{n\in\omega} \text{PSPACE}_n = \bigcup_{n\in\omega} \text{NPSPACE}_n.$$

As usual for deterministic complexity classes, PSPACE is trivially closed under complements.

Theorem VIII.5.6 (Karp [1972]) *There are* log*-space complete sets for* PSPACE*.*

Proof. Let

$$\langle x, e, n\rangle \in \mathcal{K}_0^{\text{PSPACE}} \Leftrightarrow M_e \text{ accepts } x \text{ in space } |n|$$

and proceed as in the proofs of VIII.1.18 and VIII.2.20. □

Notice that $\mathcal{K}_0^{\text{PSPACE}}$ belongs to PSPACE_1 by definition. However, there is no contradiction with the Hierarchy Theorem VIII.5.5 for PSPACE, because PSPACE_1 is *not* closed under log-space reductions: although a log-space computable function f uses only logarithmic *working space*, to check whether $f(x)$ is in $\mathcal{K}_0^{\text{PSPACE}}$ requires space proportional to $|f(x)|$, and the latter is only bounded by a polynomial in $|x|$. What the Hierarchy Theorem shows is that there can be no fixed polynomial bound of the computation space of reductions of sets in PSPACE to $\mathcal{K}_0^{\text{PSPACE}}$ (or, more generally, there can be no fixed bound of the complexity of both a log-space complete set for PSPACE and the reductions of sets in PSPACE to it). What the computational simplicity of $\mathcal{K}_0^{\text{PSPACE}}$ shows is that the complexity of a set $B \in \text{PSPACE}$ (measured by the least n such that $B \in \text{PSPACE}_n$) is mirrored in the best complexity of its reductions to $\mathcal{K}_0^{\text{PSPACE}}$.

Examples of real-life problems that turn out to be log-space complete for PSPACE are given in Karp [1972], Stockmeyer and Meyer [1973], Stockmeyer [1974], Garey and Jonhson [1979], Hopcroft and Ullman [1979], Wagner and Wechsung [1986], Johnson [1990] and Papadimitriou [1994]. Two significant examples are:

- *Intuitionistic Propositional Provability*, i.e. provability of formulas of the Intuitionistic Propositional Calculus (Statman [1979])

- *Classical Quantified Propositional Provability*, i.e. provability of formulas of the Classical Propositional Calculus with quantification over propositional letters (Stockmeyer and Meyer [1973]).

Notice that, since Boolean Satisfiability is log-space complete for NP, Classical Propositional Provability is instead log-space complete for co-NP.

Polynomial time again $\star$

The next observation describes the trivial relationships among the basic time and space complexity classes introduced so far.

Proposition VIII.5.7 (Cook [1971a], Savitch [1970])

$$\text{LOGSPACE} \subseteq \text{NLOGSPACE} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PSPACE}.$$

Proof. Of the first three inclusions, the only nontrivial one is the second, which was proved in VIII.2.22. The last inclusion follows trivially from VIII.5.3, since NP $\subseteq$ NPSPACE $=$ PSPACE. $\square$

By restricting attention to the deterministic classes one has

$$\text{LOGSPACE} \subseteq \text{P} \subseteq \text{PSPACE}.$$

Since the two extremes are different by the Space Hierarchy Theorem VII.4.13, at least one of the two inclusions is proper, but it is not known which one.

In terms of *width* and *length* of their respective computations, the difference among LOGSPACE, P and PSPACE can be described as follows:

1. a LOGSPACE computation is *very narrow* and *short*, i.e. it takes logarithmic space and hence polynomial time

2. a P computation is *narrow* and *short*, i.e. it takes polynomial time and hence polynomial space

3. a PSPACE computation is *narrow* but possibly *very long*, i.e. it takes polynomial space and hence polyexponential time.

Thus the problem of whether the two inclusions above are proper is equivalent to the problem of whether short computations can be made very narrow, and narrow computations can be made short. The fact that the two extremes are different shows that narrow computations cannot in general be made very narrow (similarly, very long computations cannot in general be made short).

We already know that, since P = ALOGSPACE, the problem of whether the first inclusion above is proper is equivalent to the problem of whether the inclusion

$$\text{LOGSPACE} \subseteq \text{ALOGSPACE}$$

is proper, i.e. whether it is possible to simulate alternating Turing machines by usual Turing machines without space loss. Similarly, since PSPACE = AP, the problem of whether the second inclusion above is proper is equivalent to the problem of whether the inclusion

$$\text{P} \subseteq \text{AP}$$

is proper, i.e. whether it is possible to simulate alternating Turing machines by usual Turing machines with only a polynomial time loss. Positive answers would provide analogues of VII.4.6, which shows that it is possible to simulate multitape Turing machines by one-tape Turing machines with no space loss, and only a quadratic time loss.

Between LOGSPACE and PSPACE there are two infinite hierarchies:

$$\begin{array}{lllllll} & \text{LOGSPACE} & \subset & \text{LOGSPACE}^2 & \subset & \cdots & \subset & \text{POLYLOGSPACE} \\ \subset & \text{PSPACE}_1 & \subset & \text{PSPACE}_2 & \subset & \cdots & \subset & \text{PSPACE}. \end{array}$$

The two hierarchies are in a sense opposite w.r.t. their relationship with P, in the following sense.

First, P is not known to be different from any class in the first hierarchy (i.e. LOGSPACE^n), while by VIII.2.26 it is different from their union (i.e. POLYLOGSPACE). We prove below that P is different from every class in the second hierarchy (i.e. PSPACE_n), while it is not known to be different from their union (i.e. PSPACE).

Second, the first class in the first hierarchy (i.e. LOGSPACE) is contained in P, but P is different from the union of the whole hierarchy (i.e. POLYLOGSPACE). We now prove that if the first class of the second hierarchy (i.e. PSPACE_1) were contained in P, then P would be equal to the union of the whole hierarchy (i.e. PSPACE).

Proposition VIII.5.8 (Book [1972])

1. *If* $\text{PSPACE}_1 \subseteq \text{P}$*, then* $\text{P} = \text{PSPACE}$*.*

2. *If* $\text{NPSPACE}_1 \subseteq \text{NP}$*, then* $\text{NP} = \text{PSPACE}$*.*

Proof. Part 1 follows by induction from

$$\text{PSPACE}_n \subseteq \text{P} \Rightarrow \text{PSPACE}_{2n} \subseteq \text{P}.$$

This is proved by noting that $|x|^{2n} = (|x|^2)^n$ and thus, by quadratic padding, one can stretch a set computed in space $|x|^{2n}$ to a set computed in space $|x|^n$ (see VIII.6.6 for a similar argument).

The proof of Part 2 is similar. □

Proposition VIII.5.9 (Book [1972]) *For every $n \geq 1$, the following holds:*

1. $\mathrm{P} \neq \mathrm{PSPACE}_n$
2. $\mathrm{P} \neq \mathrm{NPSPACE}_n$
3. $\mathrm{NP} \neq \mathrm{PSPACE}_n$
4. $\mathrm{NP} \neq \mathrm{NPSPACE}_n$.

Proof. To prove Part 1, suppose $\mathrm{P} = \mathrm{PSPACE}_n$ for some $n \geq 1$. We reach a contradiction by proving

$$\mathrm{P} \subseteq \mathrm{PSPACE}_n \subset \mathrm{PSPACE}_{n+1} \subseteq \mathrm{PSPACE}_{\,2n} \subseteq \mathrm{P}.$$

The strict inclusion comes from the Hierarchy Theorem VIII.5.5 for PSPACE; the last inclusion follows from the proof of VIII.5.8.1, because $\mathrm{PSPACE}_n \subseteq \mathrm{P}$ by half of the hypothesis; the first inclusion is the other half of the hypothesis.

The proofs of Parts 2–4 are similar. □

A similar argument does not suffice to show that $\mathrm{P} \neq \mathrm{LOGSPACE}^n$, for the following reason: quadratic padding only turns a set computed in space $\log |x|^2 = 2 \cdot \log |x|$ to a set computed in space $\log |x|$, and it thus provides only a linear speed-up.

The Polynomial Time Hierarchy again

The fact that $\mathrm{NP} \subseteq \mathrm{PSPACE}$ can be improved as follows.

Proposition VIII.5.10 (Stockmeyer [1976]) PSPACE *is closed under (polynomially) bounded quantification.*

Proof. To prove closure under existential bounded quantification, let

$$x \in A \;\Leftrightarrow\; (\exists y)_{|y| \leq p(|x|)} R(x, y),$$

where p is a polynomial and R is polynomial space computable. To check whether $R(x, y)$ holds for a fixed y requires at most polynomial space. But space can be reused, and thus the same polynomial space can be used to check $R(x, y)$ for all y such that $|y| \leq p(|x|)$. Then A is computable in polynomial space.

Closure under universal bounded quantification follows from closure under negation and existential bounded quantification. □

Corollary VIII.5.11 PH $\subseteq$ PSPACE.

Proof. PH is generated inductively by polynomially bounded quantification, from polynomial time computable matrices. Such matrices are in PSPACE because P $\subseteq$ PSPACE. The result follows by closure of PSPACE under polynomially bounded quantification. $\square$

For the next result, we use the canonical Σ_n^{P}-complete sets $\emptyset_{\mathrm{P}}^{(n)}$ defined in VIII.4.10 (by iterated relativizations of the complete set $\mathcal{K}_0^{\mathrm{NP}}$ for NP).

Theorem VIII.5.12 (Stockmeyer and Meyer [1973]) *The set* $\emptyset_{\mathrm{P}}^{(\omega)}$ *defined by*

$$\langle x, y\rangle \in \emptyset_{\mathrm{P}}^{(\omega)} \Leftrightarrow x \in \emptyset_{\mathrm{P}}^{(y)}$$

is polynomial time m-complete for PSPACE.

Proof. The proof of VIII.5.11 shows that $\emptyset_{\mathrm{P}}^{(n)} \in$ PSPACE uniformly in n, and thus that $\emptyset_{\mathrm{P}}^{(\omega)} \in$ PSPACE.

For the converse, we modify the proof of Savitch's Theorem VII.4.20 as follows. Suppose A is accepted by a deterministic Turing machine M working in polynomial space p. Since M works in space $p(|x|) \geq \log |x|$, there are only $2^{a \cdot p(|x|)}$ possible different configurations on input x, for some a (because, by VII.4.5, there are only

$$d^{p(|x|)} = (2^{\log d})^{p(|x|)} = 2^{(\log d) \cdot p(|x|)}$$

such possible configurations, for some d). By possibly modifying M in an inessential way, we can suppose that on any input x there are unique initial and final configurations c_0^x and c_f^x: after M reaches any accepting configuration, its work can be continued by erasing the working tape, moving the input head to a fixed position (e.g. to the rightmost symbol of the input), and then entering a unique special final state.

Suppose c_1 and c_2 are two configurations of M: we let

$$c_1 \vdash_n c_2 \Leftrightarrow c_2 \text{ can be obtained from } c_1 \text{ in at most } 2^n \text{ moves of } M.$$

To determine whether $x \in A$ is enough to determine whether

$$c_0^x \vdash_{a \cdot p(|x|)} c_f^x,$$

and we are thus reduced to the problem of showing that this expression is polynomial time m-reducible to $\emptyset_{\mathrm{P}}^{(\omega)}$.

We can try to proceed inductively, by using the relation

$$c_1 \vdash_{n+1} c_2 \Leftrightarrow (\exists c)(c_1 \vdash_n c \wedge c \vdash_n c_2).$$

Then each level adds only one existential quantifier (thus keeping the complexity independent of n), but it roughly doubles the length (which thus grows exponentially in n). In particular, this does not provide a polynomial time m-reduction.

However, we can use instead the relation

$$\begin{aligned} &c_1 \vdash_{n+1} c_2 \Leftrightarrow \\ &\qquad (\exists c)(\forall d)(\forall e)[(d = c_1 \wedge e = c) \vee (d = c \wedge e = c_2) \Rightarrow d \vdash_n e]. \end{aligned}$$

Then each level adds an alternation of quantifiers (thus making the complexity dependent on n), but only increases the length by a constant factor (which thus grows linearly in n).

Each of the quantifiers is polynomially bounded, because it ranges over the set of configurations of a machine working in polynomial space. Since two blocks of alternated quantifiers are added at each level, for a fixed x the expression $c_0^x \vdash_{a \cdot p(|x|)} c_f^x$ is polynomial time m-reducible to $\emptyset_{\mathrm{P}}^{(2 \cdot a \cdot p(|x|))}$, uniformly in x. Then $A \leq_m^{\mathrm{P}} \emptyset_{\mathrm{P}}^{(\omega)}$. $\square$

Since the sets $\emptyset_{\mathrm{P}}^{(n)}$ have been defined by iterated relativizations of $\mathcal{K}_0^{\mathrm{NP}}$, and thus by a polynomial time version of the jump operator (V.1.5), the set $\emptyset_{\mathrm{P}}^{(\omega)}$ is a polynomial time version of the ω-jump of the empty set (V.1.10). Thus the previous result shows that PSPACE *is the ω-level of the Polynomial Time Hierarchy*.

Corollary VIII.5.13 Second Collapse Theorem for PH (Wrathall [1976]) *If* PH = PSPACE, *then* PH $= \Delta_n^{\mathrm{P}}$ *for some n.*

Proof. If PH = PSPACE, then $\emptyset_{\mathrm{P}}^{(\omega)} \in \Sigma_n^{\mathrm{P}}$ for some n, and then all $\emptyset_{\mathrm{P}}^{(y)}$ are in Σ_n^{P}, in particular $\emptyset_{\mathrm{P}}^{(n+1)}$. But such a set is $\Sigma_{n+1}^{\mathrm{P}}$-complete, and then $\Sigma_n^{\mathrm{P}} = \Sigma_{n+1}^{\mathrm{P}}$. Thus PH $= \Delta_n^{\mathrm{P}}$ by the First Collapse Theorem VIII.4.11. $\square$

The corollary can be reformulated by saying that *if* PH *does not collapse, then it is different from* PSPACE.

The converse of the Second Collapse Theorem for PH is not known to hold. Notice that it is not automatic: if PH $= \Delta_n^{\mathrm{P}}$, then $\emptyset_{\mathrm{P}}^{(y)} \in \Delta_n^{\mathrm{P}}$ for every y, but this does not imply $\emptyset_{\mathrm{P}}^{(\omega)} \in \Delta_n^{\mathrm{P}}$, unless $\emptyset_{\mathrm{P}}^{(y)} \in \Delta_n^{\mathrm{P}}$ holds *uniformly* in y.

Relativizations

The next definition is the analogue of VIII.3.27.

Definition VIII.5.14 PSPACE^A *is the class of sets computable in deterministic polynomial space (in the length of the input) by a Turing machine with oracle A.*

Obviously,

$$\mathrm{P}^A \subseteq \mathrm{NP}^A \subseteq \mathrm{PH}^A \subseteq \mathrm{PSPACE}^A$$

for any oracle A.

Proposition VIII.5.15 (Baker, Gill and Solovay [1975]) *There is a (recursive) oracle A such that $\mathrm{P}^A = \mathrm{NP}^A = \mathrm{PH}^A = \mathrm{PSPACE}^A$.*

Proof. As in VIII.3.28, this can proved by building a set A which is polynomial time T-complete for PSPACE^A. Even more simply, it is enough here to let A be a polynomial time T-complete set for PSPACE. Then

$$\mathrm{P}^A \subseteq \mathrm{NP}^A \subseteq \mathrm{PH}^A \subseteq \mathrm{PSPACE}^A \subseteq \mathrm{PSPACE} \subseteq \mathrm{P}^A.$$

The second last inclusion follows from the fact that $A \in$ PSPACE, and PSPACE is closed under polynomial time Turing reducibility. The last inclusion holds because A is polynomial time T-complete for PSPACE. □

Turning to inequalities, we notice the following:

- Since there is an oracle A such that $\mathrm{P}^A \neq \mathrm{NP}^A$ (see VIII.3.29), *there is an oracle A such that $\mathrm{P}^A \neq \mathrm{PSPACE}^A$.*
- Since there is an oracle B such that $\mathrm{NP}^B \neq \text{co-NP}^B$ (see VIII.3.32.b), *there is an oracle B such that $\mathrm{NP}^B \neq \mathrm{PSPACE}^B$.*
- Since there is an oracle C such that PH^C does not collapse (see p. 232), by relativization of VIII.5.13 it follows that *there is an oracle C such that $\mathrm{PH}^C \neq \mathrm{PSPACE}^C$.*

The existence of oracles A_n collapsing PH^{A_n} to level n but not before (see p. 232) has been refined by giving two such oracles, respectively making level n in one case equal to PSPACE^{A_n}, and in the other case different from it (Ko [1988]). In particular, *there is an oracle A such that*

$$\mathrm{P}^A = \mathrm{PH}^A \neq \mathrm{PSPACE}^A.$$

This shows that *a positive solution to* P = NP *would not automatically imply a positive solution to* P = PSPACE, and also that the converse of the Second Collapse Theorem for PH does not relativize.

Exercise VIII.5.16 *There is an oracle A such that* $\mathrm{P}^A \neq \mathrm{NP}^A = \mathrm{PSPACE}^A$. (Dekhtiar [1976]) (Hint: diagonalize to make $\mathrm{P}^A \neq \mathrm{NP}^A$, and code a complete set for PSPACE^A into NP^A to make $\mathrm{NP}^A = \mathrm{PSPACE}^A$.)

We turn now to positive relativizations, and provide examples of restrictions both of the access to the oracle, and of the class of oracles.

Definition VIII.5.17 Bounded-Query Version of PSPACE^A (Book [1981]) PSPACE_b^A *is the class of sets computable in polynomial space (in the length of the input) and at most polynomially many queries to the oracle, by a deterministic Turing machine with oracle A.*[6]

Note that without the restriction, a query can be made at each step of a computation, and by VII.4.5 a halting computation working in space $p(|x|)$ can run for $c^{p(|x|)}$ many steps, and thus ask up to $c^{p(|x|)} \leq 2^{2^{|x|}}$ many queries.

Theorem VIII.5.18 First Positive Relativization of P = PSPACE (Selman, Xu and Book [1983], Balcázar, Book and Schöning [1985])

$$\mathrm{P} = \mathrm{PSPACE} \Leftrightarrow (\forall A)(\mathrm{P}^A = \mathrm{PSPACE}_b^A).$$

Proof. One direction is trivial, since $\mathrm{PSPACE}_b = \mathrm{PSPACE}$. The converse follows from the equality

$$\mathrm{PSPACE}_b^A = \mathrm{P}^{A \oplus \mathcal{K}_0^{\mathrm{PSPACE}}},$$

since if P = PSPACE, then $\mathcal{K}_0^{\mathrm{PSPACE}} \in \mathrm{P}$, and thus $\mathrm{PSPACE}_b^A = \mathrm{P}^A$. We now prove the equality.

In one direction, notice that polynomial time oracle computations can ask only polynomially many questions to the oracle. In particular, a set in $\mathrm{P}^{A \oplus \mathcal{K}_0^{\mathrm{PSPACE}}}$ is computable in polynomial time asking only polynomially many questions to $A \oplus \mathcal{K}_0^{\mathrm{PSPACE}}$. The questions to A are thus polynomially many, and the questions to $\mathcal{K}_0^{\mathrm{PSPACE}}$ can be answered in polynomial space. Then $\mathrm{P}^{A \oplus \mathcal{K}_0^{\mathrm{PSPACE}}} \subseteq \mathrm{PSPACE}_b^A$.

For the converse, a polynomial space computation with oracle A that queries the oracle only polynomially many times can be split into a polynomial number of phases, between successive queries. Each query can be asked to the oracle A; and each phase, being an unrelativized polynomial space computation, can be reduced to a question to the complete set $\mathcal{K}_0^{\mathrm{PSPACE}}$ for SPACE. Thus the whole computation becomes a polynomial time computation relative to the oracle $A \oplus \mathcal{K}_0^{\mathrm{PSPACE}}$, and $\mathrm{PSPACE}_b^A \subseteq \mathrm{P}^{A \oplus \mathcal{K}_0^{\mathrm{PSPACE}}}$. □

[6]In the literature, PSPACE_b^A is sometimes called PQUERY^A.

Exercises VIII.5.19 Positive relativization of NP = PSPACE. (Book [1981]) $\mathbf{NPSPACE}_b^A$ is the class of sets acceptable in polynomial space (in the length of the input) and at most polynomially many queries to the oracle, by a nondeterministic Turing machine with oracle A.[7]

a) NP = PSPACE *if and only if* $(\forall A)(\mathrm{NP}^A = \mathrm{NPSPACE}_b^A)$. (Hint: similar to VIII.5.18.)

b) *For every oracle A,* $\mathrm{PSPACE}_b^A \subseteq \mathrm{NPSPACE}_b^A \subseteq \mathrm{PSPACE}^A$.

c) $\mathrm{PSPACE}_b = \mathrm{NPSPACE}_b = \mathrm{PSPACE}$.

d) *There is an oracle B such that* $\mathrm{PSPACE}_b^B \neq \mathrm{NPSPACE}_b^B$. Thus a relativized analogue of Savitch's Theorem VII.4.20 fails in this context. (Hint: let

$$x \in A \iff (\exists y)(|y| = |x| \wedge y \in B).$$

Then $A \in \mathrm{NPSPACE}_b^B$, and one can force $A \notin \mathrm{PSPACE}_b^B$ as in VIII.3.29.)

e) *There is an oracle B such that* $\mathrm{NPSPACE}_b^B \neq \mathrm{PSPACE}^B$. (Hint: let A be as in part d). Then $A \in \mathrm{NPSPACE}_b^B$, and one can force $\overline{A} \notin \mathrm{NPSPACE}_b^B$ as in VIII.3.29. Then $\mathrm{NPSPACE}_b^B$ is not closed under complements, but PSPACE^B is.)

For other results about PSPACE_b^A and $\mathrm{NPSPACE}_b^A$, in particular for a positive relativization of PH = PSPACE, see Book and Wrathall [1981], Book, Wilson and Xu [1982], Xu, Doner and Book [1983], Book, Long and Selman [1984].

Theorem VIII.5.20 Second Positive Relativization of PH = PSPACE (Long and Selman [1986])

$$\mathrm{PH} = \mathrm{PSPACE} \iff (\forall \textit{sparse}\, A)(\mathrm{PH}^A = \mathrm{PSPACE}^A).$$

Proof. One direction is trivial, since $\emptyset$ is sparse. The converse is a corollary of VIII.5.12 and VIII.4.14, as follows.

Suppose PH = PSPACE. Then $\emptyset_{\mathrm{P}}^{(\omega)} \in \Sigma_n^{\mathrm{P}}$ for some n, and this provides a reduction of Σ_m^{P} to Σ_n^{P}, uniformly in $m \geq n$. The argument of the proof of VIII.4.14 shows that, for A sparse and $m \geq 2$, the complete set for $\Sigma_m^{\mathrm{P},A}$ is in $\Sigma_n^{\mathrm{P},A}$, again uniformly in $m \geq n$. By relativization of VIII.5.12, it then follows that the complete set for PSPACE^A is in $\Sigma_n^{\mathrm{P},A}$, and thus $\mathrm{PH}^A = \mathrm{PSPACE}^A$. □

Balcázar, Book and Schöning [1986] have strengthened the trivial direction of the last result by showing that *if* $\mathrm{PH}^A = \mathrm{PSPACE}^A$ *for some sparse A, then* PH = PSPACE. Thus one also has

$$\mathrm{PH} \neq \mathrm{PSPACE} \iff (\forall \textit{sparse}\, A)(\mathrm{PH}^A \neq \mathrm{PSPACE}^A).$$

[7] In the literature, $\mathrm{NPSPACE}_b^A$ is sometimes called $\mathrm{NPQUERY}^A$.

A look inside PSPACE $\star$

The various classes of the Polynomial Time Hierarchy are defined in terms of *nondeterministic* Turing machines, which accept an input if and only if at least one computation accepts it (see VII.4.18). More explicitly, an input is accepted if some computation accepts it, while it is rejected if no computation accepts it.

The notion of a nondeterministic Turing machine can be generalized in at least two different ways. On the one hand, a *probabilistic* Turing machine is a usual nondeterministic Turing machine with a probabilistic notion of acceptance. On the other hand, a *statistical* Turing machine is a special nondeterministic Turing machine with a preassigned proportion of accepting and rejecting computations. Gill [1977] has introduced various classes of sets based on notions of probabilistic and statistical Turing machines, as follows.

RP (for *random polynomial time*) is the class of sets accepted in polynomial time by statistical Turing machines such that, on any input, either at least half of the computations accept it, or all computations reject it. Since an output 1 is absolutely correct, while an output 0 is correct only with a probability of at least $\frac{1}{2}$, RP is a natural formalization of the notion of a *feasible Monte Carlo algorithm*, i.e. of a feasible randomized algorithm which always gives an answer, but it may lie.

ZPP (for *zero-error probability polynomial time*) is the class RP $\cap$ co-RP, which is introduced because RP is not obviously closed under complements, due to its asymmetric definition. A set in ZPP admits a pair of RP algorithms for it and its complements, which give a correct answer if one of the output is 1, and no answer if both outputs are 0 (the outputs cannot be both 1). Thus ZPP is a natural formalization of the notion of a *feasible Las Vegas algorithm*, i.e. of a feasible randomized algorithm which never lies, but it may not answer.

BPP (for *bounded-error probability polynomial time*) is the class of sets accepted in polynomial time by statistical Turing machines such that, on any input, either at least $\frac{1}{2}+\epsilon$ of the computations accept it, or at least $\frac{1}{2}+\epsilon$ reject it, where ϵ is a fixed positive number (BPP is independent of ϵ, for any $0<\epsilon<\frac{1}{2}$).

PP (for *probabilistic polynomial time*) is the class of sets accepted in polynomial time by probabilistic Turing machines which accept or reject an input according to whether more than half of the computations accept it or reject it (in case of a tie there is no answer, but this condition is inessential).

While none of the classes ZPP, RP and BPP is known to have complete sets, due to the difficult of enumerating the appropriate class of statistical Turing machines, a complete set for PP can be defined as in VIII.3.9, i.e.

$$\langle x,e,n,y\rangle \in \mathcal{K}_0^{\mathrm{PP}} \quad \Leftrightarrow \quad \text{at least } y \text{ computations of } N_e \text{ accept } x \text{ in time } |n|.$$

The relationships among the various classes above are the following:

$$\mathrm{P} \subseteq \mathrm{ZPP} \subseteq \mathrm{RP} \subseteq \mathrm{BPP} \subseteq \mathrm{PP} \subseteq \mathrm{PSPACE}.$$

None of the previous inclusions is known to be proper, since any would imply $\mathrm{P} \neq \mathrm{PSPACE}$, and there are various oracles w.r.t. which they fail (Rackoff [1982], Geske and Grollmann [1986]). The relationships of the various classes above with the Polynomial Time Hierarchy are the following:

- $\mathrm{RP} \subseteq \Sigma_1^{\mathrm{P}}$, and actually $\mathrm{RP} \subseteq \mathbf{L}_2^{\mathrm{P}}$ (Zachos and Heller [1986])
- $\mathrm{BPP} \subseteq \Delta_2^{\mathrm{P}}$ (Lautemann [1983], Sipser [1983]), but there are oracles w.r.t. which $\mathrm{BPP} \subseteq \Sigma_1^{\mathrm{P}}$ fails (Stockmeyer [1985])
- if $\mathrm{PP} \subseteq \mathrm{PH}$, then PH collapses (Toda [1989], [1991]).

In the opposite direction:

- $\Sigma_1^{\mathrm{P}} \subseteq \mathrm{PP}$
- if $\Sigma_1^{\mathrm{P}} \subseteq \mathrm{BPP}$, then PH collapses (Zachos [1983])
- $\mathrm{PH} \subseteq \mathrm{P}^{\mathcal{K}_0^{\mathrm{PP}}}$, i.e. PH is contained in the downward closure of PP w.r.t. polynomial time T-reducibility (Toda [1989], [1991]).

By using a probabilistic version of *alternating* Turing machines, in which the universal mode is replaced by a random mode, one reaches the full power of PSPACE. More precisely, the alternated versions ABPP and APP of BPP and PP turn out to be equal to PSPACE (Papadimitriou [1985]). Similarly, the related class IP of sets acceptable (with a notion of acceptance similar to BPP) by *interactive proof systems*, in which a prover exchanges information with a polynomial time bounded probabilistic verifier, is also equal to PSPACE (Lund, Fortnow, Karloff and Nisan [1990], Shamir [1990]). Interestingly, the equality $\mathrm{IP} = \mathrm{PSPACE}$ does not relativize (Aiello, Goldwasser and Håstad [1986], Fortnow and Sipser [1988]).

For surveys of complexity classes inside PSPACE and expositions of IP see Johnson [1990], Babai [1990] and Hartmanis, Chang, Ranjan and Rohatgi [1990]. For treatments, see Schöning [1986], Balcázar, Díaz and Gabarró [1988], [1990], and Papadimitriou [1994].

VIII.6 Exponential Time and Space

We have warned the reader that Sections 3–5 are hypothetical, since the classes studied there are not known to be different from P. It is now time to enter again the world of reality, and continue our study with the consideration of new classes. By the Time and Space Hierarchy Theorems VII.4.15 and VII.4.13, this can be achieved by sufficiently increasing the time or space bounds. A natural choice for a function growing faster than all polynomials is the exponential function, and thus we now consider exponential and other related bounds, for both time and space.

Exponential time

Since the simulation results VII.4.6.2 and VII.4.7.b provide quadratic trade-offs for the time bounds on one-tape and multitape Turing machines, to obtain a class independent of the model of computation we have to consider a family of bounds closed under squaring. Since $(2^{n\cdot|x|})^2 = 2^{2n\cdot|x|}$, the family $\{2^{n\cdot|x|}\}_{n\in\omega}$ would do.

Definition VIII.6.1 **EXP** *is the class of sets computable in deterministic exponential time* $2^{n\cdot|x|}$*, for some* n*.*

NEXP *is the class of sets acceptable in nondeterministic exponential time* $2^{n\cdot|x|}$*, for some* n*.*[8]

Complementing similar machine-theoretical characterizations of CONSPACE, LOGSPACE, P and PSPACE, Chandra and Stockmeyer [1976] and Ladner, Lipton and Stockmeyer [1978] have proved that EXP is the class of sets acceptable by two-way *alternating pushdown automata*. See Ladner [1981] for an exposition of this result and of its applications.

If one defines the classes

$$\mathrm{EXP}_n = \mathrm{TIME}\,[\,2^{n\cdot|x|}\,],$$

which are not independent of the model of computation, then one immediately has a hierarchy theorem.

Theorem VIII.6.2 Hierarchy Theorem for EXP.

1. $\mathrm{EXP}_n \subset \mathrm{EXP}_{n+1}$*, for each* n
2. $\mathrm{EXP} = \bigcup_{n\in\omega} \mathrm{EXP}_n$.

Proof. Part 1 follows from the Time Hierarchy Theorem VII.4.15. Part 2 is just a restatement of the definition of EXP. □

[8]In the literature, EXP and NEXP are sometimes called DEXT and NEXT, or ETIME and NETIME, or E and NE.

If one defines the classes

$$\mathrm{NEXP}_n = \mathrm{NTIME}\,[\,2^{n\cdot|x|}\,],$$

which are not independent of the model of computation, then one obtains a Hierarchy Theorem similar to VIII.6.2. In particular,

$$\mathrm{NEXP} = \bigcup_{n\in\omega} \mathrm{NEXP}_n$$

is just a restatement of the definition of NEXP, while the fact that the hierarchy is proper has been proved by Cook [1973].

Since EXP is a deterministic class, it is trivially closed under complements. However, it is not known whether NEXP is closed under complements.

Theorem VIII.6.3 (Stockmeyer and Meyer [1973]) *There are* log-*space complete sets for* EXP*. Similarly for* NEXP*.*

Proof. The proof for EXP is like the one for P (VIII.2.20), by letting

$$\langle e, x, n\rangle \in \mathcal{K}_0^{\mathrm{EXP}} \;\Leftrightarrow\; M_e \text{ accepts } x \text{ in time } 2^{|n|}.$$

The proof for NEXP is like the one for NP (VIII.3.9), by letting

$$\langle e, x, n\rangle \in \mathcal{K}_0^{\mathrm{NEXP}} \;\Leftrightarrow\; N_e \text{ accepts } x \text{ in time } 2^{|n|}. \quad \Box$$

Examples of real-life problems that turn out to be log-space complete for EXP or NEXP are given in Wagner and Wechsung [1986], and Johnson [1990].

We now to turn to a comparison of EXP and NEXP to other complexity classes. First, it follows from VII.4.25.2 that

$$\mathrm{NLINSPACE} \subseteq \mathrm{EXP},$$

but it is not known whether the inclusion is proper. Second,

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{NEXP},$$

and Book [1972] has proved that the last inclusion is proper. Between the two extremes there is an infinite hierarchy:

$$\mathrm{P} \subset \mathrm{EXP}_1 \subset \mathrm{EXP}_2 \subset \cdots \subset \mathrm{EXP} \subseteq \mathrm{NEXP},$$

and the last inclusion is not known to be proper. No inclusion is known to hold between NP and either EXP_n (for any $n \geq 1$) or EXP.

Proposition VIII.6.4 (Book [1972]) *If* $\mathrm{EXP}_1 \subseteq \mathrm{NP}$*, then* $\mathrm{EXP} \subseteq \mathrm{NP}$*.*

Proof. The result follows by induction from

$$\mathrm{EXP}_n \subseteq \mathrm{NP} \;\Rightarrow\; \mathrm{EXP}_{2n} \subseteq \mathrm{NP}.$$

This is proved by noting that $2^{2n\cdot|x|} = ((2^{|x|})^2)^n \approx (x^2)^n$ and thus, by quadratic padding of x, one can stretch a set computed in space $2^{2n\cdot|x|}$ to a set computed in space $2^{n\cdot|x|}$ (see VIII.6.6 for a similar argument). □

Proposition VIII.6.5 (Book [1972])

1. $\mathrm{NP} \neq \mathrm{EXP}_n$, *for every* $n \geq 1$
2. $\mathrm{NP} \neq \mathrm{EXP}$.

Proof. Part 1 follows as in VIII.5.9. More precisely, suppose $\mathrm{NP} = \mathrm{EXP}_n$ for some $n \geq 1$. We reach a contradiction by proving

$$\mathrm{NP} \subseteq \mathrm{EXP}_n \subset \mathrm{EXP}_{n+1} \subseteq \mathrm{EXP}_{2n} \subseteq \mathrm{NP}.$$

The strict inclusion comes from the Hierarchy Theorem VIII.6.2 for EXP; the last inclusion holds by the proof of VIII.6.4, because $\mathrm{EXP}_n \subseteq \mathrm{NP}$ by half of the hypothesis; the first inclusion is the other half of the hypothesis.

To prove Part 2, suppose $\mathrm{NP} = \mathrm{EXP}$. We reach a contradiction by proving that, for some n,

$$\mathrm{NP} \subseteq \mathrm{EXP}_n \subset \mathrm{EXP}_{n+1} \subseteq \mathrm{NP}.$$

The strict inclusion comes from the Hierarchy Theorem VIII.6.2 for EXP, and the last inclusion comes from half of the hypothesis, namely $\mathrm{EXP} \subseteq \mathrm{NP}$. The first inclusion comes from the other half of the hypothesis, as follows.

If $\mathrm{NP} \subseteq \mathrm{EXP}$, then $\mathcal{K}_0^{\mathrm{NP}} \in \mathrm{EXP}$, and thus $\mathcal{K}_0^{\mathrm{NP}}$ is computable in time $2^{m\cdot|x|}$, for some m. Moreover, every NP set is reducible to $\mathcal{K}_0^{\mathrm{NP}}$ in polynomial time, and hence in time $2^{|x|}$. Then every NP set is computable in time $2^{|x|} + 2^{m\cdot|2^{|x|}|}$, hence (since $|2^{|x|}| = |x| + 1$) in time $2^{n\cdot|x|}$ for some n, and $\mathrm{NP} \subseteq \mathrm{EXP}_n$. □

Notice that we have proved $\mathrm{NP} \neq \mathrm{EXP}$ without deciding whether any of the two classes is contained in the other. Indeed, we have already noted that neither of the two inclusions is known to hold. The next result shows that the assertion that $\mathrm{NP} \not\subseteq \mathrm{EXP}$ is stronger than $\mathrm{P} \neq \mathrm{NP}$.

Theorem VIII.6.6 First Upward Translation Theorem (Fagin [1974], Book [1974], Hartmanis and Hunt [1974], Jones and Selman [1974])

$$\mathrm{P} = \mathrm{NP} \;\Rightarrow\; \mathrm{EXP} = \mathrm{NEXP} \;\Rightarrow\; \mathrm{NP} \subseteq \mathrm{EXP}.$$

Proof. The second implication is trivial, since NP $\subseteq$ NEXP. The idea to prove the first is to increase the length of the input by an exponential factor, so that the complexity of the computation decreases by a logarithmic factor, thus stepping down from exponential to polynomial.

Suppose EXP $\neq$ NEXP and $A \in$ NEXP $-$ EXP. Consider the set B defined by

$$2^x \in B \;\Leftrightarrow\; x \in A.$$

We first prove that $B \in$ NP. Given z, to decide whether $z \in B$ first we see if z is of the form 2^x for some x; this can be done in real time, by checking whether the first digit is 1 and all the following ones are 0. If not, then $z \notin B$. If so, then $z \in B$ if and only if $|z| - 1 \in A$. But $A \in$ NEXP, and thus it is accepted by a nondeterministic Turing machine working in time exponential in the length of $|z|$, and hence polynomial in $|z|$ itself, because (for $y = |z|$)

$$2^{n\cdot|y|} = (2^{|y|})^n = (2 \cdot 2^{|y|-1})^n \leq (2 \cdot y)^n.$$

We now prove that $B \notin$ P. Otherwise, $A \in$ EXP, as follows: to decide whether $x \in A$ one sees whether $2^x \in B$, and if $B \in$ P this can be done in time polynomial in $|2^x| = x + 1 \leq 2^{|x|}$, i.e. in time exponential in $|x|$. $\square$

The result just proved holds quite tightly, in the sense that hypotheses slightly weaker than P = NP do not imply EXP = NEXP. See VIII.6.9 for an example.

Exercises VIII.6.7 Relativizations.

a) *There is an oracle A such that* $\mathrm{P}^A \neq \mathrm{NP}^A$ *and* $\mathrm{EXP}^A = \mathrm{NEXP}^A$. (Book, Wilson and Xu [1982], Kurtz [1985]) (Hint: as in VIII.3.31, make $\mathrm{P}^A \neq \mathrm{NP}^A$ by diagonalization, and $\mathrm{NEXP}^A \subseteq \mathrm{EXP}^A$ by coding a complete set for NEXP^A into EXP^A.)

b) *There is an oracle B such that* $\mathrm{EXP}^B \neq \mathrm{NEXP}^B$. (Dekhtiar [1976]) (Hint: build a set $A \in \mathrm{NEXP}^B - \mathrm{EXP}^B$, as in VIII.3.29. This also follows directly from VIII.3.29, whose proof produces a sparse set in $\mathrm{NP}^B - \mathrm{P}^B$, by relativization of VIII.6.8.)

c) *There is an oracle such that* $\mathrm{NP}^A \subset \mathrm{EXP}^A$. (Dekhtiar [1976]) (Hint: from VIII.3.28 and VIII.6.6, by VII.4.15.)

d) *There is an oracle B such that* $\mathrm{EXP}^B \subset \mathrm{NP}^B$. (Dekhtiar [1976], Gasarch and Homer [1983]) (Hint: instead of producing $A \in \mathrm{NEXP}^B - \mathrm{EXP}^B$ as in part b), spread A so that $A \in \mathrm{NP}^B - \mathrm{EXP}^B$. Also, ensure $\mathrm{EXP}^B \subseteq \mathrm{NP}^B$ by coding a complete set for EXP^B into NP^B.) See the comments after VIII.6.12 for an alternative proof.

e) *There is an oracle C such that* $\mathrm{NP}^C \not\subseteq \mathrm{EXP}^C$ *and* $\mathrm{EXP}^C \not\subseteq \mathrm{NP}^C$. (Dekhtiar [1976])

For further results see Lischke [1986], [1987], [1989], [1990].

The next result shows a connection between the problems $\mathrm{P} = \mathrm{NP}$ and $\mathrm{EXP} = \mathrm{NEXP}$ of a kind different from that given by the First Upward Translation Theorem VIII.6.6.

Proposition VIII.6.8 (Book [1974a], Hartmanis [1983]) *There are sparse sets in* $\mathrm{NP} - \mathrm{P}$ *if and only if* $\mathrm{EXP} \neq \mathrm{NEXP}$.

Proof. If $A \in \mathrm{NEXP} - \mathrm{EXP}$, then the set B defined by

$$2^x \in B \;\Leftrightarrow\; x \in A$$

is sparse by definition, and in $\mathrm{NP} - \mathrm{P}$ by the proof of VIII.6.6.

Conversely, suppose $\mathrm{EXP} = \mathrm{NEXP}$. Given a sparse set $A \in \mathrm{NP}$, we show it is in P. First, the set

$$\{\langle x, n\rangle : |A \cap \{0, \ldots, 2^x - 1\}| \geq n\}$$

is in NEXP, as follows: guess n different numbers $< 2^x$, i.e. of length $\leq x$, and see whether they are all in A. Since $A \in \mathrm{NP}$, this can be done in nondeterministic time polynomial in $x < 2^{|x|}$, and hence exponential in $|x|$ (since $(2^{|x|})^m = 2^{m\cdot|x|}$).

Since A is sparse, we know that $|A \cap \{0, \ldots, 2^x - 1\}|$ is bounded by a polynomial in $x < 2^{|x|}$, and thus by an exponential in $|x|$. We can then compute in nondeterministic exponential time (in $|x|$) first $|A \cap \{0, \ldots, 2^x - 1\}|$, by checking successive values of it, and then $A \cap \{0, \ldots, 2^x - 1\}$, by guessing the right number of elements and checking whether they are all in A.

Thus $A \cap \{0, \ldots, 2^x - 1\}$ is computable in nondeterministic exponential time in $|x|$, and hence in deterministic exponential time in $|x|$ because, by hypothesis, $\mathrm{EXP} = \mathrm{NEXP}$. But exponential in $|x|$ means polynomial in x: then $A \cap \{0, \ldots, x - 1\}$ is computable in deterministic polynomial time in $|x|$, i.e. A is in P. $\Box$

The result just proved shows that if $\mathrm{P} \neq \mathrm{NP}$ but $\mathrm{EXP} = \mathrm{NEXP}$, then no sparse set exists in $\mathrm{NP} - \mathrm{P}$, and thus any sparse set of nondeterministic polynomial time computations can be uniformly simulated in deterministic polynomial time. Thus $\mathrm{P} = \mathrm{NP}$ holds locally (on any sparse set), although it fails *globally*. If instead $\mathrm{EXP} \neq \mathrm{NEXP}$, then there are sparse sets in $\mathrm{NP} - \mathrm{P}$, and $\mathrm{P} = \mathrm{NP}$ also fails *locally* (on some sparse set).

Exercise VIII.6.9 *If* NP *contains all sets computable in deterministic time* $|x|^{\log|x|}$, *i.e. if* $\mathrm{TIME}[\,|x|^{\log|x|}\,] \subseteq \mathrm{NP}$, *then* $\mathrm{EXP} \neq \mathrm{NEXP}$. (Hartmanis, Immerman and Sewelson [1983]) (Hint: the construction of the Time Hierarchy Theorem VII.4.15 can be extended, by considering only a sparse set of witnesses for the diagonalization conditions, to produce a sparse set computable in deterministic time $|x|^{\log|x|}$ but not in polynomial time. Under the hypothesis such a set is in NP, and then there is a sparse set in NP $-$ P.)

EXP provides the smallest complexity class for which absolute degree-theoretical results have been obtained. In particular, the proof of VIII.2.36 shows that *the polynomial time m-degrees and T-degrees of* EXP *sets are not elementarily equivalent*. Downey and Nies [1997] have proved that *the first-order theories of both structures are undecidable*, and have conjectured that they have the same degree as the theory of First-Order Arithmetic.

Further results on EXP and NEXP can be found in Cook [1971], Ladner, Lynch and Selman [1975], Berman [1976], Monien [1977], Ko and Moore [1981], Selman [1982], [1982a], Watanabe [1987], Ganesan and Homer [1989], Ambos-Spies, Homer and Soare [1990], Buhrman, Homer and Torenvliet [1991], Buhrman and Homer [1992], Buhrman and Torenvliet [1994].

Beyond exponential time ⋆

We have noticed above that it is not known whether NP $\subseteq$ EXP. We thus introduce an extension of EXP for which the inclusion holds automatically.

Definition VIII.6.10 POLYEXP *is the class of sets computable in deterministic polyexponential time* $2^{|x|^n}$*, for some n.*

POLYNEXP *is the class of sets acceptable in nondeterministic polyexponential time* $2^{|x|^n}$*, for some n.*[9]

Complementing similar machine-theoretical characterizations of CONSPACE, LOGSPACE, P, PSPACE and EXP, Ladner, Lipton and Stockmeyer [1978] have proved that POLYEXP is the class of sets acceptable by two-way *multihead alternating pushdown automata*.

POLYEXP is defined in terms of bounded *time* computations. Chandra, Kozen and Stockmeyer [1981] have characterized in terms of bounded *space* computations, by showing that POLYEXP = APSPACE, i.e. POLYEXP is the class of sets computable in polynomial space in the length of the input by alternating Turing machines.

Simon [1977] has instead characterized POLYEXP as the class of sets definable by one existential quantification over polynomially bounded sets, followed by a predicate in the Polynomial Time Hierarchy. From this point of view, POLYEXP is thus a higher order analogue of NP. More precisely, if NP is a

[9]In the literature, POLYEXP and POLYNEXP are sometimes called EXPTIME and NEXPTIME, or EXP and NEXP.

polynomial time analogue of Σ_1^0, then POLYEXP is a polynomial time analogue of Σ_1^1.

If one defines the classes

$$\text{POLYEXP}_n = \text{TIME}\,[\,2^{|x|^n}\,],$$

then one immediately has a hierarchy theorem.

Theorem VIII.6.11 Hierarchy Theorem for POLYEXP.

1. $\text{POLYEXP}_n \subset \text{POLYEXP}_{n+1}$, *for each* n
2. $\text{POLYEXP} = \bigcup_{n\in\omega} \text{POLYEXP}_n$.

Proof. Part 1 follows from the Time Hierarchy Theorem VII.4.15. Part 2 is just a restatement of the definition of POLYEXP. $\Box$

Notice that $\{\text{EXP}_n\}_{n\in\omega}$ stratifies the second level of the POLYEXP Hierarchy, since $\text{EXP}_1 = \text{POLYEXP}_1$ and $\text{EXP} \subset \text{POLYEXP}_2$ (the inclusion is proper by the Time Hierarchy Theorem VII.4.15). Obviously, any level of the POLYEXP Hierarchy can be similarly stratified.

If one defines the classes

$$\text{POLYNEXP}_n = \text{NTIME}\,[\,2^{|x|^n}\,],$$

then one obtains a hierarchy theorem similar to VIII.6.11. In particular,

$$\text{POLYNEXP} = \bigcup_{n\in\omega} \text{POLYNEXP}_n$$

is just a restatement of the definition of POLYNEXP, while the fact that the hierarchy is proper has been proved by Cook [1973].

Since POLYEXP is a deterministic class, it is trivially closed under complements. However, it is not known whether POLYNEXP it is closed under complements.

Theorem VIII.6.12 *There are* log*-space complete sets for* POLYEXP. *Similarly for* POLYNEXP.

Proof. The set $\mathcal{K}_0^{\text{EXP}}$ defined in VIII.6.3 and complete for EXP is already automatically complete also for POLYEXP. The only difference in the two cases is that a reduction to $\mathcal{K}_0^{\text{EXP}}$ of a set in EXP takes the form $\langle x, e, 2^{n\cdot|x|}\rangle$, while a reduction of a set in POLYEXP takes the form $\langle x, e, 2^{|x|^n}\rangle$. The proof of VIII.2.20 shows that these reductions are all log-space computable.

Similarly, the set $\mathcal{K}_0^{\text{NEXP}}$ defined in VIII.6.3 and complete for NEXP is already automatically complete also for POLYNEXP. $\Box$

Exercises VIII.6.13 a) *Any* log*-space complete set for* EXP *is also* log*-space complete for* POLYEXP. (Hint: by a padding argument, as in VIII.6.6.)

b) *Any* log*-space complete set for* NEXP *is also* log*-space complete for* POLYNEXP.

Notice that EXP *and* NEXP *are not closed under* log*-space reductions*. A proof follows from the next result, which is suggested by the following considerations. Suppose $A \in \mathrm{EXP}$ and

$$x \in B \Leftrightarrow f(x) \in A,$$

where f is log-space computable. Since $A \in \mathrm{EXP}$, to check whether $f(x) \in A$ requires time $2^{n \cdot |f(x)|}$ for some n. Although f uses only logarithmic *working space*, to compute $f(x)$ requires polynomial *time*. Thus $|f(x)|$ is only bounded by a polynomial in $|x|$ and B is only in POLYEXP.

Proposition VIII.6.14 POLYEXP *and* POLYNEXP *are the downward closures of* EXP *and* NEXP *w.r.t. polynomial time T-reducibility. In other words, if*

$$\mathrm{P}^{\mathrm{EXP}} = \bigcup_{A \in \mathrm{EXP}} P^A \qquad \textit{and} \qquad \mathrm{P}^{\mathrm{NEXP}} = \bigcup_{A \in \mathrm{NEXP}} P^A,$$

then

$$\mathrm{POLYEXP} = \mathrm{P}^{\mathrm{EXP}} \qquad \textit{and} \qquad \mathrm{POLYNEXP} = \mathrm{P}^{\mathrm{NEXP}}.$$

Proof. The inclusion $\mathrm{POLYEXP} \subseteq \mathrm{P}^{\mathrm{EXP}}$ follows from the fact that any POLYNEXP set is log-space m-reducible to $\mathcal{K}_0^{\mathrm{EXP}}$, i.e. to an EXP set, by VIII.6.12.

For the opposite inclusion, we show that if $B \leq_T^{\mathrm{P}} A$ and $A \in \mathrm{EXP}$, then $A \in \mathrm{POLYEXP}$. Suppose B is computable from A in time p and A is computable in time $2^{n \cdot |x|}$, for some p and n. Since $B(x)$ can be computed from A in time $p(|x|)$, any query to the oracle A has length bounded by $p(|x|)$ and can be answered in time $2^{n \cdot p(|x|)} \leq 2^{p(|x|)+1}$. Thus A is computable in polyexponential time.

The proof of the second equality is similar. $\square$

We notice two consequences of the previous result. First,

$$\mathrm{EXP} = \mathrm{NEXP} \Rightarrow \mathrm{POLYEXP} = \mathrm{POLYNEXP}.$$

Second, it is impossible to find an oracle A such that $\mathrm{NP}^A = \mathrm{EXP}^A$ (which would contradict the relativization of VIII.6.5.2), by the usual method of coding a polynomial time T-complete set for EXP^A into NP^A. Otherwise, this would force all of $\mathrm{POLYEXP}^A$ into NP^A, thus making $\mathrm{EXP}^A \subset \mathrm{NP}^A$. This observation provides an alternative proof of VIII.6.7.d.

Recall that we do not know whether NP is contained in EXP. The next result implies that it is contained in POLYEXP instead.

Proposition VIII.6.15 PSPACE $\subseteq$ POLYEXP $\subseteq$ POLYNEXP.

Proof. The second inclusion is obvious. For the first inclusion, the Space-Time Trade-Off Theorem VII.4.17 shows that a set computable in space $p(|x|)$ is computable in deterministic time $c^{p(|x|)}$ for some constant c, and hence in time $2^{|x|^n}$. $\square$

None of the two inclusions is known to be proper. Since POLYEXP = APSPACE, the problem of whether the first inclusion is proper is equivalent to the problem of whether the inclusion

$$\mathrm{PSPACE} \subseteq \mathrm{APSPACE},$$

is proper, i.e. whether it is possible to simulate alternating Turing machines by usual Turing machines with only a polynomial space loss.

Exercises VIII.6.16 Relativizations. (Dekhtiar [1976]) Obviously, for any A,

$$\mathrm{P}^A \subseteq \mathrm{NP}^A \subseteq \mathrm{PSPACE}^A \subseteq \mathrm{POLYEXP}^A \subseteq \mathrm{POLYNEXP}^A.$$

a) *For any oracle A,* $\mathrm{P}^A \neq \mathrm{POLYEXP}^A$ *and* $\mathrm{NP}^A \neq \mathrm{POLYNEXP}^A$.

b) *There is an oracle A such that* $\mathrm{P}^A \neq \mathrm{NP}^A = \mathrm{POLYEXP}^A$. (Hint: code a complete set for $\mathrm{POLYEXP}^A$ into NP^A, and use part a).)

c) *There is an oracle B such that* $\mathrm{P}^B = \mathrm{PSPACE}^B \neq \mathrm{POLYEXP}^B$. (Hint: from VIII.5.15 and part a).)

d) *There is an oracle C such that* $\mathrm{POLYEXP}^C = \mathrm{POLYNEXP}^C$.

e) *There is an oracle D such that* $\mathrm{POLYEXP}^D \neq \mathrm{POLYNEXP}^D$.

Hemachandra [1989] has proved that POLYNEXP *is closed under (polynomially) bounded quantification*. In particular, the analogue of the Polynomial Time Hierarchy over NEXP (with POLYNEXP matrices and polynomially bounded quantifiers) collapses to the first level. See Schöning [1990] for a simple proof.

Babai, Fortnow and Lund [1990] have proved an analogue of IP = PSPACE, by showing that the class MIP of sets acceptable by *interactive proofs systems with multiple provers* is equal to POLYNEXP.

Exponential space

The next definition is the space analogue of VIII.6.1.

Definition VIII.6.17 EXPSPACE *is the class of sets computable in deterministic exponential space* $2^{n\cdot|x|}$*, for some n.*

NEXPSPACE *is the class of sets acceptable in deterministic exponential space* $2^{n\cdot|x|}$*, for some n.*

The next result is the analogue of VIII.5.3 and is proved in the same way.

Proposition VIII.6.18 (Savitch [1970]) EXPSPACE = NEXPSPACE.

Proof. Savitch's Theorem VII.4.20 provides a quadratic trade-off between deterministic and nondeterministic space bounds, and $(2^{n\cdot|x|})^2 = 2^{2n\cdot|x|}$. □

EXPSPACE is defined in terms of bounded *space* computations. Chandra, Kozen and Stockmeyer [1981] have characterized it in terms of bounded *time* computations, by showing that EXPSPACE = AEXP, i.e. EXPSPACE is the class of sets computable in exponential time in the length of the input by alternating Turing machines.

If one defines the classes

$$\text{EXPSPACE}_n = \{\text{sets computable in deterministic space } 2^{n\cdot|x|}\}$$

and

$$\text{NEXPSPACE}_n = \{\text{sets acceptable in nondeterministic space } 2^{n\cdot|x|}\},$$

then one immediately has a double hierarchy theorem.

Theorem VIII.6.19 Hierarchy Theorem for EXPSPACE.

1. $\text{EXPSPACE}_n \subset \text{EXPSPACE}_{n+1}$, *for each n*
2. $\text{NEXPSPACE}_n \subset \text{NEXPSPACE}_{n+1}$, *for each n*
3. $\text{EXPSPACE} = \bigcup_{n\in\omega} \text{EXPSPACE}_n = \bigcup_{n\in\omega} \text{NEXPSPACE}_n$.

Proof. Parts 1 and 2 follow from the Space Hierarchy Theorems VII.4.13 and VII.4.24, respectively.

The first equality of Part 3 is simply a restatement of the definition of EXPSPACE. The second equality follows from the first and VIII.6.18, since it is a restatement of the definition of NEXPSPACE. □

It is not known whether $\mathrm{EXPSPACE}_n = \mathrm{NEXPSPACE}_n$ for some $n \geq 1$. Notice that this might fail for every $n \geq 1$, despite the fact that

$$\bigcup_{n\in\omega} \mathrm{EXPSPACE}_n = \bigcup_{n\in\omega} \mathrm{NEXPSPACE}_n .$$

Since EXPSPACE is a deterministic class, it is trivially closed under complements.

Theorem VIII.6.20 (Stockmeyer and Meyer [1973]) *There are* log-*space complete sets for* EXPSPACE.

Proof. The proof of VIII.5.6 goes through unchanged, by letting

$$\langle e, x, n\rangle \in \mathcal{K}_0^{\mathrm{EXPSPACE}} \Leftrightarrow M_e \text{ accepts } x \text{ in space } 2^{|n|}. \quad \square$$

Examples of real-life problems that turn out to be log-space complete for EXPSPACE are in Hopcroft and Ullman [1979], Wagner and Wechsung [1986], and Johnson [1990].

One obviously has

$$\mathrm{P} \subseteq \mathrm{PSPACE} \subset \mathrm{EXPSPACE}$$

and

$$\mathrm{P} \subset \mathrm{EXP} \subseteq \mathrm{NEXP} \subseteq \mathrm{EXPSPACE},$$

where the strict inclusions follow from the Space and Time Hierarchy Theorems VII.4.13 and VII.4.15, and the last inclusion from VII.4.25.1. No inclusion is known to hold between PSPACE and either EXP or NEXP.

Proposition VIII.6.21 (Book [1974])

1. PSPACE $\neq$ EXP

2. PSPACE $\neq$ NEXP.

Proof. As in VIII.6.5.2, using the Hierarchy Theorem for PSPACE (VIII.5.5), and the existence of log-space complete sets for EXP and NEXP (VIII.6.3). $\square$

The next result is the analogue of VIII.6.6.

Theorem VIII.6.22 Second Upward Translation Theorem (Book [1974])

$$\mathrm{P} = \mathrm{PSPACE} \Rightarrow \mathrm{EXP} = \mathrm{EXPSPACE} \Rightarrow \mathrm{PSPACE} \subseteq \mathrm{EXP}$$

and

$$\mathrm{NP} = \mathrm{PSPACE} \Rightarrow \mathrm{NEXP} = \mathrm{EXPSPACE} \Rightarrow \mathrm{PSPACE} \subseteq \mathrm{NEXP}.$$

Proof. As in VIII.6.6. □

The Second Upward Translation Theorem holds tightly, in the same sense already explained for the First Upward Translation Theorem.

Exercises VIII.6.23 a) *If* PSPACE *contains all sets computable in deterministic time* $|x|^{\log|x|}$, *i.e. if* $\mathrm{TIME}[\,|x|^{\log|x|}\,] \subseteq \mathrm{PSPACE}$, *then* $\mathrm{EXP} \neq \mathrm{EXPSPACE}$. (Hartmanis, Immerman and Sewelson [1983]) (Hint: as in VIII.6.9.)

b) *If* PSPACE *contains all sets acceptable in nondeterministic time* $|x|^{\log|x|}$, *i.e. if* $\mathrm{NTIME}[\,|x|^{\log|x|}\,] \subseteq \mathrm{PSPACE}$, *then* $\mathrm{NEXP} \neq \mathrm{EXPSPACE}$. (Li [1985]) (Hint: in time $|x|^{\log|x|}$ one can uniformly enumerate the nondeterministic Turing machines working in polynomial time. Under the hypothesis, such an enumeration is in PSPACE, and hence by usual deterministic diagonalization one can obtain a set in PSPACE − NP. By exponential padding as in VIII.6.6, one obtains a set in EXPSPACE − NEXP.)

Superexponential bounds ⋆

The classes EXP, NEXP and EXPSPACE can be generalized to superexponential bounds in an obvious way, by iterating the exponential function as follows:

$$\exp_0(x) = x \qquad \text{and} \qquad \exp_{n+1} = 2^{\exp_n(x)}.$$

Thus $\exp_1(x) = 2^x$ and $\exp_n(x) = 2^{(2^{(2^{(\cdots x)})})}$ is the result of n iterations of the exponential function in base 2 on x.

Definition VIII.6.24 EXP^n *is the class of sets computable in deterministic superexponential time* $\exp_n(m \cdot |x|)$, *for some* m.

NEXP^n *is the class of sets acceptable in nondeterministic superexponential time* $\exp_n(m \cdot |x|)$, *for some* m.

EXPSPACE^n *is the class of sets computable in deterministic superexponential space* $\exp_n(m \cdot |x|)$, *for some* m.

The classes EXP, NEXP and EXPSPACE are special cases of those just defined (for $n = 1$), and a number of results proved for them can be extended to EXP^n, NEXP^n and EXPSPACE^n for every $n \geq 1$.

From the Time Hierarchy Theorem VII.4.15, one has

$$\mathrm{P} \subset \mathrm{EXP}^1 \subset \mathrm{EXP}^2 \subset \cdots,$$

and

$$\mathrm{EXP}^1 \subset \mathrm{POLYEXP} \subset \mathrm{EXP}^2,$$

and Cook [1973] has proved that

$$\mathrm{NP} \subset \mathrm{NEXP}^1 \subset \mathrm{NEXP}^2 \subset \cdots.$$

The First Upward Translation Theorem VIII.6.6 can be generalized, with a similar proof, to

$$\mathrm{P} = \mathrm{NP} \;\Rightarrow\; \mathrm{EXP}^n = \mathrm{NEXP}^n, \text{ for every } n \geq 1$$

and

$$\mathrm{EXP}^{n_0} = \mathrm{NEXP}^{n_0} \;\Rightarrow\; \mathrm{EXP}^n = \mathrm{NEXP}^n, \text{ for every } n \geq n_0.$$

Thus the problem of the relationship between deterministic and nondeterministic time bounds becomes the determination of whether P = NP and, if not, of the first $n \geq 1$ such that $\mathrm{EXP}^n = \mathrm{NEXP}^n$. Of course, it is possible that for every $n \geq 1$, $\mathrm{EXP}^n \neq \mathrm{NEXP}^n$.

Exercises VIII.6.25 a) *There is an oracle A such that* $\mathrm{EXP}^{n,A} = \mathrm{NEXP}^{n,A}$ *for every* $n \geq 1$. (Hint: by VIII.3.28.)

b) *There is an oracle A such that* $\mathrm{P}^A \neq \mathrm{NP}^A$ *and* $\mathrm{EXP}^A = \mathrm{NEXP}^A$. (Book, Wilson and Xu [1982], Kurtz [1985]) (Hint: by VIII.6.7.a.)

c) *For every $n \geq 1$ there is an oracle B_n such that* $\mathrm{EXP}^{n,B_n} \neq \mathrm{NEXP}^{n,B_n}$ *and* $\mathrm{EXP}^{n+1,B_n} = \mathrm{NEXP}^{n+1,B_n}$. (Hint: similar to part b).)

d) *There is an oracle B such that* $\mathrm{EXP}^{n,B} \neq \mathrm{NEXP}^{n,B}$ *for every* $n \geq 1$. (Sewelson [1983]) (Hint: by superexponential stretching, as in VIII.6.9 one sees that if $\mathrm{TIME}\,[\,|x|^{\log|x|}\,] \subseteq \mathrm{NP}$, then $\mathrm{EXP}^n \neq \mathrm{NEXP}^n$ for every $n \geq 1$. It is thus enough to build B such that NP^B contains the class of all sets computable with oracle B in deterministic time $|x|^{\log|x|}$. Let A^B be a set complete for this class, uniformly in B. It is enough to find B such that

$$x \in A^B \;\Leftrightarrow\; (\exists y)(|y| = |x|^2 \,\wedge\, x * y \in B),$$

since then $A^B \in \mathrm{NP}^B$. We build B by finite initial segments σ_s. At stage $s+1$, for any x of length s, compute $A^{\sigma_s}(x)$ in time $|x|^{\log|x|} \leq 2^{\log^2|x|}$, and put in $\overline{B}$ all elements z queried in the computation and such that $\sigma_s(z) = 0$, so that $A^{\sigma_s}(x) = A^B(x)$. Moreover, if $x \notin A^{\sigma_s}$, then put in $\overline{B}$ all elements $x * y$ with $|y| = s^2$. If instead $x \in A^{\sigma_s}$, put in B one of such elements that is not yet in it. Such an element exists because there are 2^{s^2} such elements, and the following holds: each element of length m restrains at most $2^{\log^2 m}$ elements, all elements of length m restrain at most $2^m \cdot 2^{\log^2 m}$ elements, and all elements of length $\leq s$ restrain at most $s \cdot 2^s \cdot 2^{\log^2 s} < 2^{s^2}$ elements.)

If, however, $\mathrm{EXP}^n \neq \mathrm{NEXP}^n$ and $\mathrm{EXP}^{n+1} = \mathrm{NEXP}^{n+1}$ for some n, or P $\neq$ NP and EXP = NEXP, then the problem could be refined by using a number of hierarchies similar to those considered in this section. For example, one could stratify EXP^{n+1} and NEXP^{n+1} into classes EXP^{n+1}_m and NEXP^{n+1}_m similar to the classes EXP_m and NEXP_m (see VIII.6.2), and ask if $\mathrm{EXP}^{n+1}_m = \mathrm{NEXP}^{n+1}_m$ for some $m \geq 1$. Again, it would be possible that for every $m \geq 1$, $\mathrm{EXP}^{n+1}_m \neq \mathrm{NEXP}^{n+1}_m$.

If however one found $\mathrm{EXP}_m^{n+1} \neq \mathrm{NEXP}_m^{n+1}$ and $\mathrm{EXP}_{m+1}^{n+1} = \mathrm{NEXP}_{m+1}^{n+1}$, or $\mathrm{P} \neq \mathrm{NP}$ and $\mathrm{EXP}_1 = \mathrm{NEXP}_1$, one could consider the Polynomial Time Hierarchy in the latter case, and a similar hierarchy obtained by iterating NEXP_m^{n+1} over EXP_m^{n+1} in the former: such a hierarchy would not collapse to the first level by hypothesis, and one could ask at which level, if any, it collapsed.

In the opposite direction, the next result shows that the possibility that $\mathrm{EXP}^n \neq \mathrm{NEXP}^n$ for every $n \geq 1$ is the worst that could happen.

Proposition VIII.6.26 $\bigcup_{n\in\omega} \mathrm{EXP}^n = \bigcup_{n\in\omega} \mathrm{NEXP}^n$.

Proof. VII.4.21 provides an exponential trade-off between deterministic and nondeterministic time bounds. □

Turning now to space bounds, from the Space Hierarchy Theorem VII.4.13 one has

$$\mathrm{PSPACE} \subset \mathrm{EXPSPACE}^1 \subset \mathrm{EXPSPACE}^2 \subset \cdots.$$

The Second Upward Translation Theorem VIII.6.22 can be generalized, with similar proofs, to

$$\mathrm{P} = \mathrm{PSPACE} \ \Rightarrow\ \mathrm{EXP}^n = \mathrm{EXPSPACE}^n, \text{ for every } n \geq 1$$

and

$$\mathrm{EXP}^{n_0} = \mathrm{EXPSPACE}^{n_0} \ \Rightarrow\ \mathrm{EXP}^n = \mathrm{EXPSPACE}^n, \text{ for every } n \geq n_0.$$

Thus the problem of the relationship between time and space bounds becomes the determination of whether $\mathrm{P} = \mathrm{PSPACE}$ and, if not, of the first $n \geq 1$ such that $\mathrm{EXP}^n = \mathrm{EXPSPACE}^n$. Of course, it is possible that for every $n \geq 1$, $\mathrm{EXP}^n \neq \mathrm{EXPSPACE}^n$.

Exercises VIII.6.27 a) *There is an oracle A such that* $\mathrm{EXP}^{n,A} = \mathrm{EXPSPACE}^{n,A}$ *for every* $n \geq 1$. (Hint: by VIII.5.15.)

b) *There is an oracle A such that* $\mathrm{P}^A \neq \mathrm{PSPACE}^A$ *and* $\mathrm{EXP}^A = \mathrm{EXPSPACE}^A$. (Hint: as in VIII.6.25.b.)

c) *For every n there is an oracle* B_n *such that* $\mathrm{EXP}^{n,B_n} \neq \mathrm{EXPSPACE}^{n,B_n}$ *and* $\mathrm{EXP}^{n+1,B_n} = \mathrm{EXPSPACE}^{n+1,B_n}$. (Hint: similar to part b).)

d) *There is an oracle B such that* $\mathrm{EXP}^{n,B} \neq \mathrm{EXPSPACE}^{n,B}$ *for every* $n \geq 1$. (Sewelson [1983]) (Hint: see VIII.6.25.d.)

If, however, $\mathrm{EXP}^n \neq \mathrm{EXPSPACE}^n$ and $\mathrm{EXP}^{n+1} = \mathrm{EXPSPACE}^{n+1}$ for some n, or $\mathrm{P} \neq \mathrm{PSPACE}$ and $\mathrm{EXP} = \mathrm{EXPSPACE}$, then the problem could be refined by using a number of hierarchies similar to those considered above for the time classes.

Also in this case, the possibility that $\mathrm{EXP}^n \neq \mathrm{EXPSPACE}^n$ for every $n \geq 1$ is the worst that could happen.

Proposition VIII.6.28 $\bigcup_{n\in\omega} \mathrm{EXP}^n = \bigcup_{n\in\omega} \mathrm{EXPSPACE}^n$.

Proof. VII.4.17 provides an exponential trade-off between space and time bounds. □

The set considered in VIII.6.26 and VIII.6.28 can thus be characterized in terms of the *same* deterministic or nondeterministic, time or space bounds. Moreover, the same set is obtained independently of whether the bounds are considered as functions of either $|x|$ or x, since the difference between the two is again an exponential. We have thus reached the limits of the investigations of Complexity Theory.

In the remaining sections of this chapter we consider classes of functions and sets that share the same closure properties with the set considered above. We are thus entering a new field, historically studied before the classes considered so far, with a shift of emphasis from an analysis of resource bounds to definability closure properties.

VIII.7 Elementary Functions

In the previous sections we have concentrated our attention on classes of functions and sets defined by resource bounds. We now turn to the consideration of classes of functions defined by variations of the schema of primitive recursion, and start with an analysis of how far one can reach by avoiding the full schema, and restricting oneself to a bounded version of it.

The class of functions thus obtained will turn out to be closed with respect to deterministic or nondeterministic time or space computations (see VIII.7.6), and to be the functional analogue of the class of sets considered in VIII.6.26 and VIII.6.28 (see VIII.7.10.3). The present section is thus a natural continuation of the previous section, and it provides the link between the two themes of this chapter.

Elementary functions

Sum and product are the basic arithmetic functions, and together with their inverses they generate the field of rational numbers (from 0 and 1). To obtain the real numbers, one has to complete such a field, and one possible way of doing this is by considering converging series, which can be seen either as infinitary sums, or as infinite sequences of finite sums. Indeed, the infinitary sum $\sum_{y\in\omega} x_y$ can be seen as an operation on the sequence $\{x_y\}_{y\in\omega}$, producing the sequence $\{\sum_{y\leq z} x_y\}_{z\in\omega}$. Thus the completion process producing the reals from the rationals corresponds to the introduction of a new infinitary operation uniformly producing iterated finite sums.

But a sequence $\{x_y\}_{y\in\omega}$ indexed by natural numbers is simply a function $f(x,y)=x_y$, and thus the previous infinitary operation has an obvious analogue in terms of functions on natural numbers, namely the *bounded sum* operation $\sum_{y\leq z} f(\vec{x},y)$ defined on p. I.25 by

$$\sum_{y\leq 0} f(\vec{x},y)=f(\vec{x},0) \qquad \text{and} \qquad \sum_{y\leq z+1} f(\vec{x},y)=(\sum_{y\leq z} f(\vec{x},y))+f(\vec{x},z+1).$$

This is a definition by primitive recursion, and thus the class of primitive recursive functions is closed under it. But one can take bounded sum as a primitive operation and see which functions are generated by it, starting from sum, product and their inverses.

For symmetry we also consider the analogous operation of *bounded product*, defined by

$$\prod_{y\leq 0} f(\vec{x},y)=f(\vec{x},0) \qquad \text{and} \qquad \prod_{y\leq z+1} f(\vec{x},y)=(\prod_{y\leq z} f(\vec{x},y))\cdot f(\vec{x},z+1).$$

The next class is thus a version of the real numbers or, more abstractly, of the completion of the field of rational functions over the natural numbers.

Definition VIII.7.1 (Kalmar [1943], Csillag [1947]) *The class* $\boldsymbol{\mathcal{E}}$ *of* **elementary functions** *is the smallest class of functions:*

1. *containing the initial functions,* $x+y$ *and* $x-y$
2. *closed under composition*
3. *closed under bounded sum and product.*

A predicate is elementary if its characteristic function is.

Constable [1973a] has shown that, by replacing bounded sum and product by *weak* bounded sum and product, the previous definition would provide an alternative characterization of the class PF. Thus, it is not surprising that similar closure properties and characterizations hold for the two classes (historically, such results were first proved for $\mathcal{E}$ and later extended to PF).

Closure properties

Notice that $\mathcal{E}$ contains the functions $x\cdot y$, x^y and $x!$, since

$$\begin{array}{rcll} x\cdot y &=& \sum_{z\leq y-1} f(x,z) & \text{where } f(x,z)=x \\ x^y &=& \prod_{z\leq y-1} f(x,z) & \text{where } f(x,z)=x \\ x! &=& \prod_{z\leq x-1} g(x,z) & \text{where } g(x,z)=z+1. \end{array}$$

In particular, there is no need of explicitly adding $x\cdot y$ in VIII.7.1.1. Similarly, the next result implies that $\frac{x}{y}$ can be obtained automatically.

Definition VIII.7.2 Bounded Versions of Recursion (Skolem [1923])

1. *A function f is defined from a relation R by* **bounded μ-recursion** *if*

$$f(\vec{x}, z) = \begin{cases} \mu y R(\vec{x}, y) & \textit{if } (\exists y \leq z) R(\vec{x}, y) \\ 0 & \textit{otherwise.} \end{cases}$$

   $f(\vec{x}, z)$ is usually indicated by $\mu y_{\leq z} R(\vec{x}, y)$.

2. *A function f is defined from functions g, h and t by* **bounded primitive recursion** *if*

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}) \\ f(\vec{x}, y+1) &= h(\vec{x}, y, f(\vec{x}, y)) \\ f(\vec{x}, y) &\leq t(\vec{x}, y). \end{aligned}$$

Proposition VIII.7.3 Closure Properties of the Elementary Functions and Predicates (Skolem [1923], Bereczki [1952], Kleene [1952], Grzegorczyk [1953])

1. *The elementary functions are closed under bounded primitive recursion and bounded μ-recursion.*

2. *The elementary predicates are closed under logical connectives and bounded quantifiers.*

Proof. Closure under logical connectives, bounded quantifiers and bounded μ-recursion has been proved on pp. I.25–26. In particular, the characteristic functions of $\neg P$, $P \wedge Q$ and $(\forall y \leq z) R(\vec{x}, y)$ are respectively $1 - c_P$, $c_P \cdot c_Q$ and $\prod_{y \leq z} c_R(\vec{x}, y)$. In addition,

$$\mu y_{\leq z} R(\vec{x}, y) = \sum_{y \leq z} (y \cdot g(\vec{x}, y)),$$

where

$$g(\vec{x}, y) = \text{characteristic function of } R(\vec{x}, y) \wedge (\forall z < y) \neg R(\vec{x}, z).$$

Closure under bounded primitive recursion is reduced to closure under bounded μ-recursion, in a way similar to (but simpler than) the proof of I.1.8, which reduced closure under primitive recursion to closure under μ-recursion (see note 10 on p. 268).

First, we notice that the following functions and predicates are elementary:

- *x divides y*
$$x \mid y \ \Leftrightarrow\ (\exists z \leq y)(x \cdot z = y).$$

- *the exponent of k in the decomposition of y*
$$\exp(y,k) = \mu x_{\leq y}[k^x \mid y \ \wedge\ \neg(k^{x+1} \mid y)].$$

- *x is a prime*
$$Pr(x) \ \Leftrightarrow\ x \geq 2 \ \wedge\ (\forall y \leq x)(y \mid x \ \rightarrow\ y = 1 \ \vee\ y = x).$$

- *x is the n-th prime number*
$$Pr(x,n) \ \Leftrightarrow\ Pr(x) \ \wedge\ n = \sum_{z<x} c_{Pr}(z).$$

  Notice that $c_{Pr}(z)$ is 1 if z is a prime and 0 otherwise, so that $\sum_{z<x} c_{Pr}(z)$ is the number of primes below x. The expression above thus states that x is the n-th prime if it is a prime and below it there are n primes (since the index of the first prime is 0).

- *the n-th prime number*
$$p_n = \mu x_{\leq 2^{2^n}} Pr(x,n).$$

  Notice that the bound 2^{2^n} is elementary, because $\mathcal{E}$ contains the exponential function and is closed under composition (and hence it contains all finite iterations of the exponential function).

  We still have to check, by induction on n, that the bound 2^{2^n} for the n-th prime p_n is correct. Obviously it is for $n = 0$, since $p_0 = 2 = 2^{2^0}$. Euclid's proof shows that between $p_{n-1}+1$ and $p_0 \cdots p_{n-1}+1$ there is a prime, and then by inductive hypothesis
$$\begin{array}{lllll} p_n & \leq & p_0 \cdots p_{n-1} + 1 & \leq & 2^{2^0} \cdots 2^{2^{n-1}} + 1 \\ & = & 2^{2^0+\cdots+2^{n-1}} + 1 & = & 2^{2^n-1} + 1 \qquad \leq \ 2^{2^n}. \end{array}$$

Suppose now that f is defined from elementary functions g, h and t by bounded primitive recursion as
$$\begin{array}{rcl} f(\vec{x},0) & = & g(\vec{x}) \\ f(\vec{x},y+1) & = & h(\vec{x},y,f(\vec{x},y)) \\ f(\vec{x},y) & \leq & t(\vec{x},y). \end{array}$$

We code the sequence of values of f from 0 to y, by using prime decomposition.[10] Let

$$m = p_0^{f(\vec{x},0)} \cdots p_y^{f(\vec{x},y)}.$$

Then

$$f(\vec{x}, y) = \exp(m, p_y),$$

and an estimate of m in terms of $\vec{x}$ and y can easily be obtained, using the fact that f is bounded by t, as follows. For every $z \leq y$,

$$f(\vec{x}, z) \leq t(\vec{x}, z) \leq \sum_{z \leq y} t(\vec{x}, z) \qquad \text{and} \qquad p_z \leq p_y,$$

and thus

$$m \leq \left(p_y^{\sum_{z \leq y} t(\vec{x},z)} \right)^{y+1}.$$

Let $s(\vec{x}, y)$ be the right hand side of the last inequality, which is elementary by the closure properties already proved. The graph of f

$$G_f(\vec{x}, y, z) \;\Leftrightarrow\; f(\vec{x}, y) = z$$

is also elementary, since

$$\begin{aligned} G_f(\vec{x}, y, z) \;\Leftrightarrow\; & (\exists m \leq s(\vec{x}, y))[\exp(m, p_0) = g(\vec{x}) \wedge \\ & (\forall i < y)(\exp(m, p_{i+1}) = h(\vec{x}, i, \exp(m, p_i)) \;\wedge\; z = \exp(m, p_y)]. \end{aligned}$$

Finally, f is elementary since

$$f(\vec{x}, y) = \mu z_{\leq t(\vec{x},y)} G_f(\vec{x}, y, z),$$

because t is a bound for f. $\square$

The previous result allows us to prove that many functions and predicates are elementary, for example:

$$\begin{aligned} x \leq y \quad & \Leftrightarrow \quad x - y = 0 \\ x = y \quad & \Leftrightarrow \quad x \leq y \;\wedge\; y \leq x \\ x < y \quad & \Leftrightarrow \quad x \leq y \;\wedge\; \neg(x = y) \end{aligned}$$

[10]This is the part of the proof similar to I.1.8, the differences being the following.

On the one hand, in I.1.8 only sum and product were given at the start (together with the characteristic function of equality), and thus the coding mechanism had to be defined only in terms of them (using the β-function). The present proof needs instead a (super)exponential to produce a bound for the n-th prime, which is provided by closure under bounded product (and composition).

On the other hand, in I.1.8 unbounded μ-recursion was available, and thus unbounded primitive recursion could be simulated.

and
$$\frac{x}{y} = \mu z_{\leq x}[y \cdot z \leq x < y \cdot (z+1)],$$
so that the inverse of $x \cdot y$ is also elementary.

Alternative characterizations

The next result is the analogue of VIII.2.15.

Proposition VIII.7.4 $\mathcal{E}$ as a Bounded Analogue of the Class of Primitive Recursive Functions (Grzegorczyk [1953]) *The class of elementary functions is the smallest class of functions:*

1. *containing the initial functions and x^y*
2. *closed under composition*
3. *closed under bounded primitive recursion.*

Proof. Let $\mathcal{C}$ be the smallest class satisfying the stated conditions. Clearly every function in $\mathcal{C}$ is elementary, since x^y is elementary and $\mathcal{E}$ is closed under bounded primitive recursion by VIII.7.3.1.

Conversely, we need to show that $x + y$ and $x - y$ are in $\mathcal{C}$ and that $\mathcal{C}$ is closed under bounded sum and product. First, $x + y$, $x - y$ and $x \cdot y$ can be defined in the standard way (see pp. I.24–25) by primitive recursion starting from the successor operation (which is an initial function), and thus we only need to show that such primitive recursions can be bounded by functions in $\mathcal{C}$. Indeed, all of $x + y$, $x - y$ and $x \cdot y$ are bounded by $(x + 1)^{y+1}$, which is in $\mathcal{C}$ because the exponential function and the successor operation are, and $\mathcal{C}$ is closed under composition.

Bounded sum and product can be defined by primitive recursion, as recalled at the beginning of the present section, and thus one only needs to show that they are bounded by functions in $\mathcal{C}$. Since
$$\sum_{y \leq z} f(\vec{x}, y) \leq (z+1) \cdot \max\{f(\vec{x}, y) : y \leq z\}$$
and
$$\prod_{y \leq z} f(\vec{x}, y) \leq (\max\{f(\vec{x}, y) : y \leq z\})^{z+1},$$
the problem reduces to showing that the bounded maximum is in $\mathcal{C}$. But
$$\max\{f(\vec{x}, y) : y \leq z\} = f(\vec{x}, h(\vec{x}, z)),$$

where $h(\vec{x}, z)$ is defined by primitive recursion bounded by z, as follows:

$$\begin{aligned} h(\vec{x}, 0) &= 0 \\ h(\vec{x}, z+1) &= \begin{cases} h(\vec{x}, z) & \text{if } f(\vec{x}, z+1) \le f(\vec{x}, h(\vec{x}, z)) \\ z+1 & \text{otherwise} \end{cases} \\ &= C(h(\vec{x}, z), z+1, f(\vec{x}, z+1) - f(\vec{x}, h(\vec{x}, z))), \end{aligned}$$

where

$$C(x, y, z) = \begin{cases} x & \text{if } z = 0 \\ y & \text{if } z > 0 \end{cases}$$

is defined by primitive recursion bounded by $x + y$, as follows:

$$\begin{aligned} C(x, y, 0) &= x \\ C(x, y, z+1) &= y. \quad \square \end{aligned}$$

Proposition VIII.7.5 $\mathcal{E}$ as a Bounded Analogue of the Class of Recursive Functions (Grzegorczyk [1953]) *The class of elementary functions is the smallest class of functions:*

1. *containing the initial functions, $x - y$ and x^y*
2. *closed under composition*
3. *closed under bounded μ-recursion.*

Proof. Let $\mathcal{C}$ be the smallest class satisfying the stated conditions: clearly every function in $\mathcal{C}$ is elementary, since $x - y$ and x^y are elementary, and $\mathcal{E}$ is closed under bounded μ-recursion by VIII.7.3.1.

Conversely, by VIII.7.4 it is enough to show that $\mathcal{C}$ is closed under bounded primitive recursion. We proceed by a number of steps.

- *$\mathcal{C}$ is closed under logical connectives*
  The characteristic function of $\neg P(x)$ is $0^{c_P(x)}$ (since $0^0 = 1$ and $0^1 = 0$), and the characteristic function of $P(x) \vee Q(x)$ is $(c_P(x))^{(0^{c_Q(x)})}$ (since $1^a = 1$, $0^{(0^1)} = 1$, and $0^{(0^0)} = 0$).

- *$x \le y$ and $x = y$ are in $\mathcal{C}$*
  First,

  $$x \le y \iff x - y = 0$$

  is in $\mathcal{C}$, because its characteristic function is $1 - (x - y)$. Then

  $$x = y \iff x \le y \wedge y \le x$$

  is also in $\mathcal{C}$, by closure under logical connectives.

- *$x+y$ and $x \cdot y$ are in $\mathcal{C}$*
  First,
  $$x \cdot y = \mu z_{\leq (x+1)^{(y+1)}}[(2^x)^y = 2^z]$$
  is in $\mathcal{C}$, because both $=$ and the exponential function are in $\mathcal{C}$. Then
  $$x + y = \mu z_{\leq (x+1)\cdot(y+1)}[2^x \cdot 2^y = 2^z]$$
  is also in $\mathcal{C}$.

- *$\mathcal{C}$ is closed under bounded quantifiers*
  The characteristic function of $(\exists y \leq z)R(\vec{x}, y)$ is
  $$c_R(\vec{x}, \mu y_{\leq z}[c_R(\vec{x}, y) = 1]),$$
  which is in $\mathcal{C}$ (if c_R is) because $=$ is in $\mathcal{C}$.

To prove that $\mathcal{C}$ is closed under bounded primitive recursion, we refer to the proof of VIII.7.3.1. The only change needed here is that we have to avoid the three uses of closure under bounded sum made there, namely:

- *definition of bounded μ-recursion*
  $\mathcal{C}$ is closed under bounded μ-recursion by definition.

- *definition of the predicate $Pr(x,n)$*
  We express the fact that x is the n-th prime by considering the auxiliary number
  $$y = 2^1 \cdot 3^2 \cdots x^{n+1},$$
  in whose decomposition the i-th prime appears with exponent $i+1$, and saying that x is a prime appearing in the decomposition of y with exponent $n+1$:
  $$\begin{aligned} Pr(x,n) \quad \Leftrightarrow \quad & Pr(x) \ \wedge \ (\exists y \leq (n+1)(2^{2^n})^{n+1}) \\ & \quad [\exp(y,2) = 1 \ \wedge \ \exp(y,x) = n+1 \ \wedge \\ & \quad (\forall a)_{a \leq y}(\forall b)_{a<b\leq y}(Pr(a) \ \wedge \ Pr(b) \ \wedge a|y \ \wedge \ b|y \ \Rightarrow \\ & \qquad (\exists c)_{a \leq c < b}(Pr(c) \ \wedge \ (\forall z)_{a<z<b} \neg Pr(z) \ \wedge \\ & \qquad\qquad \exp(y,b) = \exp(y,c)+1)]. \end{aligned}$$
  The bound on y, which follows from the observation (made in the proof of VIII.7.3.1) that the n-th prime can be bounded by 2^{2^n}, is in $\mathcal{C}$ by the closure properties already proved. Then $Pr(x,n)$ is in $\mathcal{C}$ by closure under bounded quantifiers and the rest of the proof of VIII.7.3.1.

- *definition of a bound of* $\{t(\vec{x},0),\ldots,t(\vec{x},y)\}$
  It is enough to prove closure of $\mathcal{C}$ under bounded maximum, which follows from closure under bounded μ-recursion and bounded quantifiers, as follows:
$$\max_{z\leq y}\{t(\vec{x},z)\} = \mu a.\,(\forall z \leq y)(t(\vec{x},z)\leq a).$$

With the modifications just discussed, the proof of VIII.7.3.1 shows that $\mathcal{C}$ is closed under bounded primitive recursion. $\square$

VIII.7.1, VIII.7.4 and VIII.7.5 show that *modulo* $x+y$, $x-y$ *and* x^y, *the bounded versions of sum and product, primitive recursion and* μ*-recursion are equivalent*. Thus $\mathcal{E}$ is at the same time an analogue of the smaller class of polynomial time computable functions, and of the bigger classes of primitive recursive and recursive functions.

Various other characterizations of $\mathcal{E}$ are considered in Grzegorczyk [1953], Skolem [1954], Rödding [1964], [1965], [1966], Parsons [1968], and Marchenkov [1980]. In particular, the latter shows that $\mathcal{E}$ *can be generated by composition alone, from finitely many appropriate initial functions*.

Computation time and space

We now turn to complexity-theoretical properties of $\mathcal{E}$, thus connecting this section to the previous ones. The next result is the paradigm for a number of results to be proved for other classes.

Theorem VIII.7.6 Ritchie-Cobham Property for $\mathcal{E}$ (Ritchie [1963], Cobham [1964]) *The following are equivalent:*

1. f *is elementary*
2. f *is computable in deterministic elementary time*
3. f *is computable in nondeterministic elementary time*
4. f *is computable in deterministic elementary space*
5. f *is computable in nondeterministic elementary space.*

Proof. Parts 2, 3, 4 and 5 are equivalent by VII.4.17, VII.4.20 and VII.4.21, since $\mathcal{E}$ is closed under exponential (because it contains x^y, and is closed under composition). Thus, it is enough to prove the equivalence between Parts 1 and 2.

- *an elementary function is computable in elementary time*

  We proceed by induction on the definition of elementary functions, by giving an outline of the procedure (the actual details depend on the model of computation used).

  The initial functions, as well as $x + y$ and $x \dot{-} y$, are actually polynomial time computable, and thus computable in elementary time (since every polynomial is elementary).

  If f has been defined by composition from $g_1, \ldots, g_m$ and h, i.e.

  $$f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x})),$$

  and T_{g_i} and T_h are the times needed to compute g_i and h, then the time T_f needed to compute f is roughly

  $$T_f(\vec{x}) = (\sum_{1 \leq i \leq n} T_{g_i}(\vec{x})) + T_h(g_1(\vec{x}), \ldots, g_m(\vec{x})),$$

  and if g_i, T_{g_i} and T_h are elementary then so is T_f, by the closure properties of $\mathcal{E}$.

  If f has been defined by bounded sum from g, i.e.

  $$\begin{array}{rcl} f(\vec{x}, 0) & = & g(\vec{x}, 0) \\ f(\vec{x}, z+1) & = & f(\vec{x}, z) + g(\vec{x}, z+1) \\ & = & h(\vec{x}, z, f(\vec{x}, z)), \end{array}$$

  where h is a composition of initial functions, $+$ and g, and T_g and T_h are the times needed to compute g and h, then the time T_f needed to compute f is roughly

  $$\begin{array}{rcl} T_f(\vec{x}, 0) & = & T_g(\vec{x}, 0) \\ T_f(\vec{x}, z+1) & = & T_f(\vec{x}, z) + T_h(\vec{x}, z, f(\vec{x}, z)). \end{array}$$

  By expanding the recursion one has

  $$T_f(\vec{x}, z) = T_g(\vec{x}, 0) + \sum_{y<z} T_h(\vec{x}, y, f(\vec{x}, y)),$$

  and if f and T_g are elementary then so are T_h (by the previous part of the proof, since h is defined by composition from functions computable in elementary time) and T_f (by the closure properties of $\mathcal{E}$).

  If f has been defined by bounded product from g, then we can proceed similarly, this time using a function h such that

  $$h(\vec{x}, z, f(\vec{x}, z)) = f(\vec{x}, z) \cdot g(\vec{x}, z+1).$$

- *a function computable in elementary time is elementary*
  Using the elementary coding procedure by prime decomposition (already used in the proof of VIII.7.3), we define an elementary function g such that

  > if a computation of the Turing machine coded by e on inputs $\vec{x}$ halts in less than z steps, then $g(e, \vec{x}, z)$ bounds the code of such a computation.

  If f is computable in time t by the Turing machine M_e, by the Improved Normal Form Theorem[11] VIII.2.8 we then have

  $$f(\vec{x}) = \mathcal{U}(\mu y_{\leq g(e,\vec{x},t(\vec{x}))} \mathcal{T}_n(e, \vec{x}, y)),$$

  for some elementary function $\mathcal{U}$ and predicates $\mathcal{T}_n$. But $\mathcal{E}$ is closed under bounded μ-recursion by VIII.7.3, and thus f is elementary if so is t.

  It remains to outline the definition of g (the actual details depend on the model of computation used).

  The code of a computation taking at most z steps has the form

  $$p_0^{a_0} \cdots p_z^{a_z} \leq p_z^{(z+1)\cdot a},$$

  where a_i codes the instantaneous configuration of the machine at step i, and $a = \max\{a_i : i \leq z\}$. An instantaneous configuration can be represented (as on p. I.49) by a sequence

  $$\sigma_1 \cdots \sigma_j \, q \, \sigma_{j+1} \cdots \sigma_m$$

  in which all the consecutive symbols σ_i (including the blanks) written in the relevant portion of the tape (at least that between the two outermost nonblank symbols) are indicated, together with the state q and the head position (shown by the position of the state symbol among the alphabet symbols).

  Now m has an elementary bound $m(\vec{x}, z)$: if the computation takes at most z steps, on the tape there can be at most z symbols, plus the number of symbols originally needed to write the inputs $\vec{x}$. Since each symbol and state has a code number less than e (because e codes all relevant information), we may suppose

  $$a \leq \sum_{n \leq m(\vec{x},z)} p_n^e,$$

[11] For the present proof we only need a version of the Normal Form Theorem with elementary function $\mathcal{U}$ and predicates $\mathcal{T}_n$. This follows immediately from the original proof of II.1.2, by the closure properties of $\mathcal{E}$.

and hence let

$$g(e,\vec{x},z) = p_z^{(z+1)\cdot\sum_{n\leq m(\vec{x},z)} p_n^e},$$

which is elementary by the closure properties of $\mathcal{E}$. $\Box$

Notice that the first part of the proof holds trivially for the polynomial time computable functions, since the polynomials are polynomial time computable. In other words, if f is polynomial time computable, then T_f is bounded by a polynomial time computable function.

Symmetrically, the second part of the proof fails badly for the polynomial time computable functions, since any such function is bounded by $2^{p(|\vec{x}|)}$ for some polynomial p (see VIII.2.16.2). In other words, if T_f is bounded by a polynomial time computable function, then f is computable in polyexponential time, but not necessarily in polynomial time.

Corollary VIII.7.7 (Meyer and McCreight [1969]) *$\mathcal{E}$ is a complexity class with respect to time, space and any measure elementarily related to them.*

Proof. $\mathcal{E}$ is obviously an r.e. set of total recursive functions. By VII.3.8, it is then enough to prove that $\mathcal{E}$ is self-bounded and has the Ritchie-Cobham property.

The latter has just been proved in VIII.7.6 for time and space (and hence for any measure elementarily related to them).

For the former, given a finite set $\mathcal{F}$ of elementary functions $\{f_1,\ldots,f_n\}$, the function f defined by

$$f(x) = \sum_{1\leq i\leq n} f_i(x)$$

is elementary (because $\mathcal{E}$ is closed under bounded sum), and it majorizes every f_i. Thus $\mathcal{E}$ is self-bounded. $\Box$

Rate of growth

In the previous subsection we have looked at the complexity of elementary functions from the point of view of the resources needed to compute them. We now take a different point of view, and look at their possible rates of growth.

The idea is simple: to define $\mathcal{E}$ we closed an initial stock of simple functions (the fastest growing of which was addition) under finite iterations of sums and products. Since the exponential function grows faster than both sum and product, every elementary function should be dominated by a finite iteration of the exponential function. The next result confirms this guess and provides an analogue of VIII.2.16.

Theorem VIII.7.8 A Skeleton for $\mathcal{E}$ (Bereczki [1952], Grzegorczyk [1953]) *Let*

$$\exp_0(x) = x \qquad \textit{and} \qquad \exp_{n+1}(x) = 2^{\exp_n(x)},$$

so that $\exp_1(x) = 2^x$*, and* $\exp_n(x) = 2^{(2^{(2^{(\cdots x)})})}$ *is the result of* n *iterations of the exponential function in base* 2 *on* x*. Then:*

1. $\exp_n$ *is elementary, for each* n
2. *every elementary function is dominated by some* $\exp_n$*, in the sense that given* f *elementary, there is an* n *such that* $f(\vec{x}) \leq \exp_n(\sum \vec{x})$*, for almost every* $\vec{x}$
3. *the diagonal function* $d_{\mathcal{E}}(x) = \exp_x(x)$ *dominates every elementary function, and thus it is not elementary.*

Proof. Before turning to the proof itself, we notice that the properties of the exponential function trivially imply the following properties of the functions $\exp_n$:

1. $x \leq \exp_n(x)$
2. if $n \leq m$, then $\exp_n(x) \leq \exp_m(x)$
3. if $x \leq y$, then $\exp_n(x) \leq \exp_n(y)$
4. $\exp_m(\exp_n(x)) = \exp_{m+n}(x)$.

We now turn to the proof.

Part 1 is obvious, since the exponential function is elementary, and $\mathcal{E}$ is closed under composition.

Part 3 follows from Part 2 and Property 2 above, since $\exp_n(x) \leq \exp_x(x)$ for every n and $x > n$.

Part 2 is proved by induction on the definition of elementary functions, as follows:

- *initial functions,* $x + y$ *and* $x \dot{-} y$
  They are obviously dominated by $\exp_1$. For example,
  $$x \dot{-} y \leq x + y \leq 2^{x+y} = \exp_1(x+y).$$
- *composition*
  If f has been defined by composition from $g_1, \ldots g_m$ and h, i.e.
  $$f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x})),$$

by induction hypothesis there is an n such that $g_i(\vec{x}) \leq \exp_n(\sum \vec{x})$ for almost every $\vec{x}$, and $h(\vec{y}) \leq \exp_n(\sum \vec{y})$ for almost every $\vec{y}$ (actually, the induction hypothesis gives different n's for $g_1, \dots, g_m$ and h, but Property 2 above shows that their maximum works for all of them simultaneously). Then

$$\begin{array}{rcll}
f(\vec{x}) & = & h(g_1(\vec{x}), \dots, g_m(\vec{x})) & \\
& \leq & \exp_n(\sum_{1\leq i\leq m} g_i(\vec{x})) & \text{by induction hypothesis on } h \\
& \leq & \exp_n(\sum_{1\leq i\leq m} \exp_n(\sum \vec{x})) & \text{by induction hypothesis on } g_i \\
& & & \text{and Property 3} \\
& \leq & \exp_n(m \cdot \exp_n(\sum \vec{x})) & \\
& \leq & \exp_n(\exp_{n+1}(\sum \vec{x})) & \text{if } \sum \vec{x} \text{ is sufficiently large} \\
& = & \exp_{2n+1}(\sum \vec{x}) & \text{by Property 4.}
\end{array}$$

- *bounded sum and product*
  If f has been defined by bounded sum from g, i.e.

$$f(\vec{x}, z) = \sum_{y \leq z} g(\vec{x}, y),$$

by induction hypothesis there is an n such that $g(\vec{x}, y) \leq \exp_n(y + \sum \vec{x})$ for almost every $\vec{x}$ and y. Then

$$\begin{array}{rcll}
f(\vec{x}, z) & = & \sum_{y\leq z} g(\vec{x}, y) & \\
& \leq & \sum_{y\leq z} \exp_n(y + \sum \vec{x}) & \text{by induction hypothesis on } g \\
& \leq & (z+1) \cdot \exp_n(z + \sum \vec{x}) & \text{by Property 3} \\
& \leq & \exp_{n+1}(z + \sum \vec{x}) & \text{if } \sum \vec{x} \text{ is sufficiently large.}
\end{array}$$

Similarly, if f has been defined by bounded product from g, i.e.

$$f(\vec{x}, z) = \prod_{y \leq z} g(\vec{x}, y),$$

by induction hypothesis there is an n such that $g(\vec{x}, y) \leq \exp_n(y + \sum \vec{x})$ for almost every $\vec{x}$ and y. Then

$$\begin{array}{rcll}
f(\vec{x}, z) & = & \prod_{y\leq z} g(\vec{x}, y) & \\
& \leq & \prod_{y\leq z} \exp_n(y + \sum \vec{x}) & \text{by induction hypothesis on } g \\
& \leq & (\exp_n(z + \sum \vec{x}))^{z+1} & \text{by Property 3} \\
& \leq & \exp_m(z + \sum \vec{x}) & \text{if } \sum \vec{x} \text{ is sufficiently large}
\end{array}$$

for an appropriate m. □

Hierarchies

By combining VIII.7.8 and VIII.7.6 we can immediately obtain hierarchies for $\mathcal{E}$. The next definition and result are the functional analogues of VIII.6.24, VIII.6.26 and VIII.6.28.

Definition VIII.7.9 $\mathbf{EXPF}^n$ *is the class of functions computable in deterministic time* $\exp_n(m \cdot |x|)$*, for some* m*.*

$\mathbf{EXPSPACEF}^n$ *is the class of functions computable in deterministic space* $\exp_n(m \cdot |x|)$*, for some* m*.*

Theorem VIII.7.10 Hierarchy Theorem for $\mathcal{E}$ (Ritchie [1963])

1. $\mathrm{EXPF}^n \subset \mathrm{EXPF}^{n+1}$*, for each* n
2. $\mathrm{EXPSPACEF}^n \subset \mathrm{EXPSPACEF}^{n+1}$*, for each* n
3. $\mathcal{E} = \bigcup_{n\in\omega} \mathrm{EXPF}^n = \bigcup_{n\in\omega} \mathrm{EXPSPACEF}^n$*.*

Proof. Parts 1 and 2 follow from the Time and Space Hierarchy Theorems VII.4.15 and VII.4.13. Part 3 follows from VIII.7.8 and VIII.7.6, as we now show.

First, by VIII.7.6, each elementary function is computable in elementary time and hence, by VIII.7.8, in time $\exp_n(x) = \exp_{n+1}(|x|)$ (since $x \leq 2^{|x|}$) for some n. Then

$$\mathcal{E} \subseteq \bigcup_{n\in\omega} \mathrm{EXPF}^n.$$

Second, by VII.4.17,

$$\bigcup_{n\in\omega} \mathrm{EXPF}^n \subseteq \bigcup_{n\in\omega} \mathrm{EXPSPACEF}^n.$$

Finally, each $\exp_n$ is elementary by VIII.7.8, and thus each function computable in space $\exp_n(m \cdot |x|)$ is elementary by VIII.7.6. Then

$$\bigcup_{n\in\omega} \mathrm{EXPSPACEF}^n \subseteq \mathcal{E}. \quad \Box$$

Although both EXPF^n and $\mathrm{EXPSPACEF}^n$ provide hierarchies for $\mathcal{E}$, we do not know whether these are distinct or coincide (see the discussion at the end of Section 6). However, we have

$$\mathrm{EXPF}^n \subseteq \mathrm{EXPSPACEF}^n \subseteq \mathrm{EXPF}^{n+1}$$

because, by VII.4.17, a function computable in space $m \cdot |x|$ is computable in time $c^{m\cdot|x|} = 2^{(\log c)\cdot m\cdot|x|}$ for some constant c.

A number of other hierarchies for $\mathcal{E}$ are possible. A local variation of the two hierarchies just considered consists of using x in place of $|x|$ in VIII.7.9 (the two definitions are related, since $2^{|x|-1} \leq x < 2^{|x|}$). A hierarchy based on a different principle is considered in VIII.8.24.

Ritchie [1963] and Herman [1969] consider space hierarchies obtained by starting with the class of functions computable by finite automata, and by taking at successor stages the class of functions computable in space belonging to the class at the previous stage (the difference between the two authors being the use of bounds in $|x|$ or x, respectively). Cleave [1963] considers a hierarchy obtained by counting the number of jump instructions used in computations on register machines (see also Kasai and Adachi [1980]). Herman [1971] shows that these hierarchies, although not coinciding, are interwoven.

We thus seem to lack a canonical hierarchy inside $\mathcal{E}$, although those considered above (based on time and space) have become standard for the classification of the complexity of elementary functions. However, the class $\mathcal{E}$ has strong invariance properties, as its various alternative characterizations show, and it is the starting point of a canonical hierarchy for the class of primitive recursive functions (see VIII.8.12).

The diagonal function ⋆

We have seen in VIII.7.8 that the diagonal function $\exp_x(x)$ leads out of the elementary functions, and it is thus a natural measure of the complexity of $\mathcal{E}$. The goal of this subsection is to analyze its rate of growth in terms of the number of steps needed to obtain it starting from the simplest function (the constant function with value 0), and iterating the simplest function that would increase the values (the successor function).

It turns out that a different diagonal function, obtained by iterating the full exponential x^y in place of 2^y, lends itself to a better analysis in these terms, and we thus use it in place of $\exp_x(x)$.

Definition VIII.7.11 *The* **canonical diagonal function for $\mathcal{E}$** *is the function* $\boldsymbol{d_{\mathcal{E}}}$ *defined by*

$$d_{\mathcal{E}}(x) = e_x(x),$$

where

$$e_0(x) = x \qquad \textit{and} \qquad e_{n+1}(x) = x^{e_n(x)}.$$

Notice that each function e_n is elementary (being an iteration of the exponential function), and grows faster than the corresponding $\exp_n$ (for $x \geq 2$). Thus the sequence $\{e_n\}_{n\in\omega}$ could be used as a skeleton for $\mathcal{E}$, in the same way as $\{\exp_n\}_{n\in\omega}$ was used in VIII.7.8, with similar consequences.

The reference for our analysis of $d_{\mathcal{E}}$ is the following sequence of functions.

Definition VIII.7.12 The Slow Growing Hierarchy (Wainer [1972]) *The sequence $\{f_\alpha\}_\alpha$ is defined by induction on the countable ordinals, as follows:*

$$\begin{array}{rcll} f_0(x) & = & 0 & \\ f_{\alpha+1}(x) & = & f_\alpha(x)+1 & \\ f_\alpha(x) & = & f_{\alpha_x}(x) & \quad \text{if } \alpha \text{ is a limit} \end{array}$$

where, for α a limit, $\{\alpha_x\}_{x\in\omega}$ is an increasing sequence of ordinals with limit α (called a **fundamental sequence** *for α).*

The Slow Growing Hierarchy is also called the Pointwise Hierarchy, since to compute f_α for the argument x, we only need values of other f_β's for the same argument.

It is quite obvious that the definition of f_α is ambiguous, since it depends on the fundamental sequences chosen for the limit ordinals (in other words, the definition requires not only ordinals, but additional structure on them). Indeed, $f_\alpha(x)$ is just the number of steps it takes to obtain to 0 by the following rules:

- start from α
- go from a successor ordinal $\beta+1$ to its predecessor β
- go from a limit ordinal β with fundamental sequence $\{\beta_x\}_{x\in\omega}$ to β_x.

Hence any strictly increasing function f can be obtained as f_ω, simply by choosing $\omega_x = f(x)$.

We can think of f_α as being a functional representation of the ordinal α. More precisely, f_α is a way of explicitly describing α in terms of a starting point, a successor operation and a countable limit operation. Thus, the problem is how to build the sequence of f_α's from below, i.e. how to build the countable ordinals inductively. This reduces to the problem of choosing natural fundamental sequences for the limit ordinals. We now see that there are, indeed, canonical choices for the fundamental sequences of small ordinals.[12]

Recall that every ordinal $\alpha \neq 0$ has a unique expansion to the base ω, of the form

$$\alpha = \omega^{\beta_1}\cdot x_1 + \cdots + \omega^{\beta_m}\cdot x_m,$$

where $x_1, \ldots, x_m$ are positive integers and $\beta_m < \cdots < \beta_1$ (the **Cantor normal form** of α). The Cantor normal form of α is called *pure* if, inductively, every β_i is finite or in pure Cantor normal form (thus a pure Cantor normal form is a term built up from ω and finite ordinals by ordinal sum, product and exponentiation).

[12]The following discussion requires a minimal background on ordinal arithmetic, which can be obtained for example from Monk [1969] or Levy [1979].

Recall also that ϵ_0 is the smallest α such that $\alpha = \omega^\alpha$. If $\alpha < \epsilon_0$, then in the Cantor normal form of α we have $\beta_1 < \alpha$, and we can use the expansion in inductive definitions. From this representation, it follows that every limit ordinal $\alpha < \epsilon_0$ can be uniquely written in one of the forms

$$\beta + \omega^{\gamma+1} \qquad \text{or} \qquad \beta + \omega^\gamma,\ \gamma \text{ a limit}$$

with $\gamma < \alpha$. This justifies the next definition.

Definition VIII.7.13 Fundamental Sequences up to ϵ_0. *If $\alpha < \epsilon_0$ is a limit ordinal, then*

$$\alpha_x = \begin{cases} \omega^\gamma \cdot x & \text{if } \alpha = \omega^{\gamma+1} \\ \omega^{\gamma_x} & \text{if } \alpha = \omega^\gamma, \text{ with } \gamma \text{ a limit} \\ \beta + \gamma_x & \text{if } \alpha = \beta + \gamma, \text{ with } \gamma \text{ a limit.} \end{cases}$$

Moreover,

$$(\epsilon_0)_0 = \omega \qquad \text{and} \qquad (\epsilon_0)_{x+1} = \omega^{(\epsilon_0)_x}.$$

In the following, we use the fundamental sequences just defined in the definition VIII.7.12 of f_α, when $\alpha \le \epsilon_0$.

Proposition VIII.7.14 The Position of $d_{\mathcal{E}}$ in the Slow Growing Hierarchy (Wainer [1972]) *For $\alpha < \epsilon_0$, $f_\alpha(x)$ is obtained by first writing α in pure Cantor normal form to the base ω, and then substituting x for ω everywhere. In particular:*

1. $e_n = f_{(\epsilon_0)_n}$
2. $d_{\mathcal{E}} = f_{\epsilon_0}$.

Proof. The result follows from the following properties:

1. $f_{\alpha+\beta}(x) = f_\alpha(x) + f_\beta(x)$, *for every* $\alpha, \beta < \epsilon_0$

   This is proved by induction on β:

   - $\beta = 0$
     There is nothing to prove.
   - $\beta = \gamma + 1$

$$\begin{array}{rcll} f_{\alpha+(\gamma+1)}(x) & = & f_{(\alpha+\gamma)+1}(x) & \text{by associativity of } + \\ & = & f_{\alpha+\gamma}(x) + 1 & \text{by definition of } f_{(\alpha+\gamma)+1} \\ & = & (f_\alpha(x) + f_\gamma(x)) + 1 & \text{by induction hypothesis} \\ & = & f_\alpha(x) + (f_\gamma(x) + 1) & \text{by associativity of } + \\ & = & f_\alpha(x) + f_{\gamma+1}(x) & \text{by definition of } f_{\gamma+1}. \end{array}$$

- *β a limit*

$$\begin{array}{rcll} f_{\alpha+\beta}(x) & = & f_{(\alpha+\beta)_x}(x) & \text{by definition of } f_{\alpha+\beta} \\ & = & f_{\alpha+\beta_x}(x) & \text{by definition of } (\alpha+\beta)_x \\ & = & f_\alpha(x) + f_{\beta_x}(x) & \text{by induction hypothesis} \\ & = & f_\alpha(x) + f_\beta(x) & \text{by definition of } f_\beta. \end{array}$$

2. *$f_{\omega^\alpha}(x) = x^{f_\alpha(x)}$, for every $\alpha < \epsilon_0$*

   This is proved by induction on α:

- $\alpha = 0$

$$\begin{array}{rcll} f_{\omega^0}(x) & = & f_1(x) & \text{by definition of } \omega^0 \\ & = & 1 & \text{by definition of } f_1 \\ & = & x^0 & \text{by definition of } x^0 \\ & = & x^{f_0(x)} & \text{by definition of } f_0. \end{array}$$

- $\alpha = \beta + 1$

$$\begin{array}{rcll} f_{\omega^{\beta+1}}(x) & = & f_{(\omega^{\beta+1})_x}(x) & \text{by definition of } f_{\omega^{\beta+1}} \\ & = & f_{\omega^\beta \cdot x}(x) & \text{by definition of } (\omega^{\beta+1})_x \\ & = & f_{\underbrace{\omega^\beta + \cdots + \omega^\beta}_{x \text{ times}}}(x) & \text{by definition of } \omega^\beta \cdot x \\ & = & \underbrace{f_{\omega^\beta}(x) + \cdots + f_{\omega^\beta}(x)}_{x \text{ times}} & \text{by Property 1 above} \\ & = & x \cdot f_{\omega^\beta}(x) & \\ & = & x \cdot x^{f_\beta(x)} & \text{by induction hypothesis} \\ & = & x^{f_\beta(x)+1} & \\ & = & x^{f_{\beta+1}(x)} & \text{by definition of } f_{\beta+1}. \end{array}$$

- *α a limit*

$$\begin{array}{rcll} f_{\omega^\alpha}(x) & = & f_{(\omega^\alpha)_x}(x) & \text{by definition of } f_{\omega^\alpha} \\ & = & f_{\omega^{\alpha_x}}(x) & \text{by definition of } (\omega^\alpha)_x \\ & = & x^{f_{\alpha_x}(x)} & \text{by induction hypothesis} \\ & = & x^{f_\alpha(x)} & \text{by definition of } f_\alpha. \quad \square \end{array}$$

Notice that the properties proved in the proof of VIII.7.14 are quite general, and hold for any system of fundamental sequences which satisfies the following rules, for β a limit:

- $\omega_x = x$
- $(\alpha + \beta)_x = \alpha + \beta_x$

- $(\alpha \cdot \beta)_x = \alpha \cdot \beta_x$
- $(\alpha^\beta)_x = \alpha^{\beta_x}$.

Then we actually have:

- $f_\omega(x) = x$
- $f_{\alpha+\beta}(x) = f_\alpha(x) + f_\beta(x)$
- $f_{\alpha\cdot\beta}(x) = f_\alpha(x) \cdot f_\beta(x)$
- $f_{\alpha^\beta}(x) = (f_\alpha(x))^{f_\beta(x)}$.

Exercise VIII.7.15 Goodstein sequences. Given n and x, to write n in **pure base** x means to write n in base x, then write the exponents of the decomposition in base x, and so on. For example, for $n = 195$ and $x = 2$,

$$195 = 2^{2^2+2^1+2^0} + 2^{2^2+2^1} + 2^1 + 2^0.$$

The **Goodstein sequence** relative to (n, x) is the sequence of numbers obtained by starting with n, and iterating the following process: write n in pure base x, change the base from x to $x + 1$, subtract 1, and start again. For example, for $n = 195$ and $x = 2$, the first two steps are

$$3^{3^3+3^1+3^0} + 3^{3^3+3^1} + 3^1$$

and

$$4^{4^4+4^1+4^0} + 4^{4^4+4^1} + \underbrace{4^0 + 4^0 + 4^0}_{3=4^1-1}.$$

Every Goodstein sequence ends in 0. (Goodstein [1944]) (Hint: given n and x, if α is obtained from the representation of n in pure base x by substituting ω for x everywhere, then $n = f_\alpha(x)$ by VIII.7.14. Changing the base to $x + 1$ means considering $f_\alpha(x+1)$. Subtracting 1 means considering $f_\alpha(x+1)-1 = f_{p(\alpha,x+1)}(x+1)$, where

$$p(\alpha, x) = \begin{cases} 0 & \text{if } \alpha = 0 \\ \beta & \text{if } \alpha = \beta + 1 \\ p(\alpha_x, x) & \text{if } \alpha \text{ is a limit.} \end{cases}$$

Thus the ordinals decrease, and since there cannot be an infinite descending sequence of ordinals, the process leads to 0.)

The statement of Goodstein's result is clearly formalizable in Peano Arithmetic $\mathcal{PA}$, but the proof given here is not (because it uses ordinals, and hence induction, up to ϵ_0). Kirby and Paris [1982] have shown that Goodstein's result is not provable in $\mathcal{PA}$. This improves the fact, proved by Gentzen [1936], that the principle of ϵ_0-induction is not provable in $\mathcal{PA}$, since Goodstein's result is (an arithmetical formulation of) a special case of ϵ_0-induction.

A recursion theoretic proof of the Kirby and Paris independence result (given by Cichon [1983]) proceeds as follows. The argument $\geq x$ in which we hit 0 when starting from $f_\alpha(x)$ is clearly $g_\alpha(x)$, where g_α is the function defined as follows (see also VIII.8.22):

$$\begin{aligned} g_0(x) &= 0 \\ g_{\alpha+1}(x) &= g_\alpha(x+1) \\ g_\alpha(x) &= g_{\alpha_x}(x) \qquad \text{if } \alpha \text{ is a limit.} \end{aligned}$$

Goodstein's result states that for every n and x there is a $y \geq x$ such that the process hits 0 at y, starting from $f_\alpha(x)$. If we could prove this in $\mathcal{PA}$, we would uniformly prove the totality of the function g_α for every $\alpha < \epsilon_0$, and hence the totality of g_{ϵ_0}. But g_{ϵ_0} is not provably total in $\mathcal{PA}$, by VIII.9.15 and VIII.9.20.2 (actually, g_{ϵ_0} is the first of the g_α's that is not provably total in $\mathcal{PA}$).

The idea of obtaining independence results of Π^0_2 formulas (which are realized by total recursive functions, by II.1.13.2) in a given system by proving that they cannot be realized by functions provably total in the system is due to Kreisel [1958a]. Other applications of this method are discussed in Smorynski [1980], [1982], [1983], and given in Ketonen and Solovay [1981] and Simpson [1987].

Relativizations

The notion of elementary function can be relativized in the usual way, following II.3.1.

Definition VIII.7.16 *If g is a total function, the class* $\boldsymbol{\mathcal{E}(g)}$ *of* **functions elementary in** $\boldsymbol{g}$ *is the smallest class of functions:*

1. *containing the initial functions, $x+y$, $x-y$ and g*
2. *closed under composition, bounded sum and bounded product.*

If A is a set, the class of **functions elementary in** $\boldsymbol{A}$ *is the class of functions elementary in c_A.*
A predicate is elementary in g or A if its characteristic function is.

The next definition is the analogue of II.3.2.

Definition VIII.7.17 *Given two functions f and g, we say that:*

1. *f is* **elementarily reducible** *to g ($f \leq_{\mathcal{E}} g$) if f is elementary in g*
2. *f is* **elementarily equivalent** *to g ($f \equiv_{\mathcal{E}} g$) if $f \leq_{\mathcal{E}} g$ and $g \leq_{\mathcal{E}} f$.*

As usual, $\leq_{\mathcal{E}}$ is a reflexive and transitive relation, and thus $\equiv_{\mathcal{E}}$ is an equivalence relation, whose equivalence classes are called **elementary degrees**. Since the structure of the elementary degrees with the partial ordering induced on them by $\leq_{\mathcal{E}}$ is similar to the structure of polynomial time degrees studied in Section 2, and no elementary difference is known, we do not pursue its study here.

Having a relativized notion of elementary function, the main result proved in VIII.7.6 can be restated as: *every recursive function is elementary in the time needed to compute it*. More concisely, $f \leq_{\mathcal{E}} T_f$.

The next result is the relativization of VIII.7.6.

Proposition VIII.7.18 (Constable [1970], Meyer and Ritchie [1972]) *If g is computable in time elementary in g, then the following are equivalent:*

1. *f is elementary in g*
2. *f is computable in time elementary in g*
3. *f is computable in space elementary in g.*

Proof. The relativization of the proof of VIII.7.6 shows that if f is elementary in g then it is computable in time elementary in the time needed to compute g ($T_f \leq_{\mathcal{E}} T_g$). Since g is computable in time elementary in g ($T_g \leq_{\mathcal{E}} g$), then so is f ($T_f \leq_{\mathcal{E}} g$, by transitivity of $\leq_{\mathcal{E}}$).

Conversely, by VIII.7.6 any function f is elementary in the time needed to compute it ($f \leq_{\mathcal{E}} T_f$). If f is computable in time elementary in g ($T_f \leq_{\mathcal{E}} g$), then f is elementary in g ($f \leq_{\mathcal{E}} g$, by transitivity of $\leq_{\mathcal{E}}$). $\square$

Corollary VIII.7.19 *If g is computable in time elementary in g, then $\mathcal{E}(g)$ is a complexity class w.r.t. time and space.*

Proof. From VIII.7.18, as in VIII.7.7. $\square$

The next exercises deal with a rich class of functions for which the hypothesis of the last result and corollary is satisfied.

Exercises VIII.7.20 Elementarily honest functions. By analogy with the notion of honesty introduced in VII.2.13, we say that a function f is **elementarily honest** if $f(x)$ is computable in time $g(x, f(x))$, for some elementary function g.

a) *A function is elementarily honest if and only if it has an elementary graph.* (Meyer and Ritchie [1972]) (Hint: by the proof of VIII.7.6, there is an elementary function h such that $f(x) = h(x, t(x))$ whenever f is computable in time t. If f is elementarily honest, there is g elementary such that $f(x)$ is computable in time $g(x, f(x))$, and thus $f(x) = h(x, g(x, f(x)))$. Then the graph of f is elementary, because

$$f(x) = z \Leftrightarrow z = h(x, g(x, z)).$$

Conversely, if the graph of f is elementary, then it is computable in elementary time t. To compute $f(x)$, search for the first z such that $f(x) = z$: each check requires time $t(x, z)$, and only $f(x)$ checks are needed. Then let $g(x, y) = \sum_{z \leq y} t(x, z)$.)

b) *The elementarily honest functions are cofinal in the recursive functions*, i.e. given a recursive function f there is an elementarily honest function g such that g dominates f everywhere. (Axt [1959]) (Hint: as in VIII.2.11.b.)

Other properties of the elementarily honest functions can be obtained from VIII.2.11, as in part b).

The next exercises provide relativized versions of VIII.7.8.

Exercises VIII.7.21 a) *Given any function g such that $g(x) \geq x$, the sequence*

$$g_0(x) = g(x) \qquad \textit{and} \qquad g_{n+1}(x) = 2^{g_n(x)}$$

dominates every function elementary in g, and its diagonal function is not elementary in g. (Hint: as in VIII.7.8.)

b) *Given any function g such that $g(x) \geq 2^x$, the sequence*

$$g_0(x) = x \qquad \textit{and} \qquad g_{n+1}(x) = g(g_n(x))$$

dominates every function elementary in g, and its diagonal function is not elementary in g. (Hint: as for part a).)

VIII.8 Primitive Recursive Functions

The class of primitive recursive functions is probably the most natural subrecursive class, and was isolated and studied well before the full class of recursive functions. For a while it was even thought that it might coincide with the class of computable functions (see Hilbert [1926]), until Sudan [1927] and Ackermann [1928] produced a computable function that is not primitive recursive (see VIII.8.10).

In this section we analyze the power of primitive recursion, and show that the only way of exceeding its strength is to work simultaneously on two or more variables, since any possible recursion on a single variable is reducible to a primitive recursion (see VIII.8.5).

Primitive recursive functions

We defined the class of primitive recursive functions at the very beginning of the book (I.1.6), and reproduce the definition here for the reader's convenience.

Definition VIII.8.1 (Dedekind [1888], Skolem [1923], Gödel [1931]) *The class $\boldsymbol{\mathcal{P}_1}$ of* **primitive recursive functions** *is the smallest class of functions:*

1. *containing the initial functions*
2. *closed under composition*
3. *closed under primitive recursion.*

A predicate is primitive recursive if its characteristic function is.

The story of the notion of primitive recursion and of the first properties of $\mathcal{P}_1$ is told in Section I.1.

Closure properties

In Section I.7 we have already proved some strong closure properties of $\mathcal{P}_1$, in particular under:

1. *course-of-value recursion* (I.7.1), in which the definition of $f(\vec{x}, y+1)$ may involve not only the previous value $f(\vec{x}, y)$, but any number of (and possibly all) the values $\{f(\vec{x}, z)\}_{z \leq y}$ already obtained;
2. *simultaneous primitive recursion* (I.7.2), in which a finite number of functions $f_1, \ldots, f_n$ are defined simultaneously, and the definition of any $f_m(\vec{x}, y+1)$ may involve not only $f_m(\vec{x}, y)$, but any number of (and possibly all) the previous values $\{f_i(\vec{x}, y)\}_{1 \leq i \leq n}$.

An obvious but useful property is the following.

Proposition VIII.8.2 (Gödel [1931]) *$\mathcal{P}_1$ is closed under elementary operations. In particular, $\mathcal{E} \subseteq \mathcal{P}_1$.*

Proof. By the characterization VIII.7.4, since bounded primitive recursion is a special case of primitive recursion. □

Exercises VIII.8.3 a) *$\mathcal{P}_1$ is closed under recursion with substitution of functions in place of parameters*, i.e. recursion of the kind

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}) \\ f(\vec{x}, y+1) &= h(\vec{x}, y, f(\vec{t}(\vec{x}), y)), \end{aligned}$$

where $\vec{t}$ and $\vec{x}$ have the same number of components, i.e. $\vec{t}(\vec{x}) = t_1(\vec{x}), \ldots, t_n(\vec{x})$ if $\vec{x} = x_1, \ldots, x_n$. (Peter [1934], Csillag [1947]) (Hint: for simplicity, let $n = 1$. The idea is that if we order the arguments of f into a doubly infinite matrix with rows corresponding to the arguments for a fixed y, the procedure to compute $f(x, y)$ requires only one argument from each previous row, and thus only a sequence of values determined by the primitive recursive function t. We are thus making a primitive recursion on a primitive recursive ordering. Formally,

$$\begin{aligned} f(x,0) &= g(x) \\ f(x,1) &= h(x,0,g(t(x))) \\ f(x,2) &= h(x,1,h(x,0,g(t^{(2)}(x)))) \\ &\cdots \\ f(x,n+1) &= h(x,n,h(x,n-1,\ldots,h(x,0,g(t^{(n+1)}(x)))\ldots)), \end{aligned}$$

and so

$$f(x,n) = f_1(x, f_2(x,n), n),$$

where

$$\begin{aligned} f_1(x,z,0) &= z \\ f_1(x,z,n+1) &= h(x,n,f_1(x,z,n)) \end{aligned}$$

and

$$f_2(x,n) = g(t^{(n)}(x)).)$$

b) $\mathcal{P}_1$ *is closed under unnested recursion on any number of variables*, where for the case of two variables such a recursion proceeds as follows:

$$\begin{aligned} f(\vec{x},y,0) &= g_1(\vec{x},y) \\ f(\vec{x},0,z+1) &= g_2(\vec{x},z), \end{aligned}$$

and $f(\vec{x},y{+}1,z{+}1)$ is obtained explicitly from known functions, using a finite number of values for arguments less than $(y+1,z+1)$ in the lexicographical order of pairs. (Peter [1936]) (Hint: the idea is that we can order the pairs of natural numbers in a sequence, with the property that to compute $f(\vec{x},y,z)$ we only need values of f for pairs preceding (y,z). It is possible to do this in a primitive recursive way because we can find the pairs preceding (y,z) ahead of time, without computing any value of f. But then f is defined by course-of-value recursion on this ordering. More precisely, define

$$\begin{aligned} z_0 &= \text{bound on } z \text{ for the pairs needed to compute } f(\vec{x},y,z'), \text{ with } z' \leq z \\ z_1 &= \text{bound on } z \text{ for the pairs needed to compute } f(\vec{x},y-1,z'), \text{ with } z' \leq z_0, \end{aligned}$$

and so on. Then the required order is

$$\cdots(y-2,0)\cdots(y-2,z_1)\ (y-1,0)\cdots(y-1,z_0)\ (y,0)\cdots(y,z).$$

In other words, if we can predict the search space in a primitive recursive way, then we only need a bounded μ-recursion on the search space, and the whole procedure is thus primitive recursive.)

The next definition captures the most general form of recursion on one variable, including in particular the possibility of *nestings*, i.e. expressions such as $f(\vec{x}, f(\vec{x},y))$.

Definition VIII.8.4 *A function f is defined from g, g_1, ..., g_n by* **nested recursion** *if*

$$\begin{aligned} f(\vec{x},0) &= g(\vec{x}) \\ f(\vec{x},y+1) &= h(\vec{x},y), \end{aligned}$$

where $h(\vec{x},y)$ is a numerical term built up from:

- *natural numbers n*
- *the variables $\vec{x}$ and y*
- *the functions $g_1,\dots,g_m$*
- *the functional letter f*
- *the function p (for 'predecessor'), defined as:*

$$p(a,b) = \begin{cases} a & \textit{if } a < b \\ 0 & \textit{otherwise.} \end{cases}$$

Moreover, $h(\vec{x},y)$ must be built in such a way that f appears in it only in contexts of the form $f(\vec{t},p(s,y+1))$, where $\vec{t}$ and s are terms.[13]

The next result shows that the class of primitive recursive functions is closed under the previous kind of recursion.

Theorem VIII.8.5 Nested Recursion on One Variable (Peter [1934]) *$\mathcal{P}_1$ is closed under nested recursion.*

Proof. We first introduce some terminology, to simplify the discussion:

- ***f*-term** is a term of the form $f(\vec{t},s)$, where $\vec{t}$ and s are terms
- **normal term** is a term in which f occurs only in the context of f-terms of the form $f(\vec{t},p(s,n))$, where $\vec{t}$ and s are terms
- **proper term** is a term not containing f-terms, i.e. without occurrences of f
- **reducible term** is a term containing an f-term of the form $f(\vec{t},s)$, where $\vec{t}$ and s are proper terms
- **constant term** is a term not containing the variables $\vec{x}$ and y.

[13] This condition expresses the fact that all values of f used in the definition of $f(\vec{x},y+1)$ are of the form $f(\vec{t},s)$, where s has a numerical value $\leq y$.

The idea to compute $f(\vec{x}, y+1)$ is obvious. We have to reduce $h(\vec{x}, y)$ to a proper term, i.e. to eliminate all occurrences of f in it, so that we can obtain the value by a direct evaluation of the known functions $g_1, \ldots, g_n$. We can eliminate the f-terms in an orderly fashion, by starting with the innermost terms. In general $h(\vec{x}, y)$ may contain many occurrences of f, and so the computation takes the form of a tree. Also, since in general there might be *nestings*, i.e. f-terms of the form $f(\vec{t}, s)$ where not all of $\vec{t}$ and s are proper, we will have to develop the tree until all branches end in proper terms, evaluate the terminal nodes, and retrace our steps back to the preceding nodes, until we can finally evaluate the root of the computation tree. This is the quite complex computation that we want to reduce to a primitive recursion.

By definition $h(\vec{x}, y)$ is a normal term, and thus we restrict our attention to normal terms. The basic problem is how to evaluate constant normal terms, which admit the following manipulations:

- *evaluation u^* of a proper constant term u*
  Since u is proper, it does not contain either variables or occurrences of f. It is then an explicit expression built up from constants and known functions, and it can thus be assigned a numerical value u^*.

- *one-step reduction of a reducible constant term u to a term u^-*
  Since u is reducible and constant, it contains an f-term $f(\vec{t}, s)$ such that $\vec{t}$ and s are proper constant terms. Then u^- is obtained by replacing in u one occurrence of $f(\vec{t}, s)$ by

$$\begin{array}{rl} g(\vec{t}) & \text{if } s^* = 0 \\ h(\vec{t}, n) & \text{if } s^* = n+1. \end{array}$$

  Notice that, since $h(\vec{x}, y)$ is normal by definition, if u is normal then so is u^-.

  Obviously, the process of reduction is not deterministic, since there might be many ways of reducing a reducible term. In the following, we will specify a particular reduction procedure.

Evaluation and reduction are the basic steps for the calculation of the numerical value of a constant normal term. We can easily mimic them in a primitive recursive way, on code numbers. Namely, let

$$\langle u \rangle = \text{a canonical code number of the term } u.$$

The following functions and predicates are then primitive recursive:

$$ev(x) \quad = \quad \begin{cases} u^* & \text{if } \langle u \rangle = x \text{ and } u \text{ is a proper constant term} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} R(x) &\Leftrightarrow u = \langle x \rangle \text{ and } u \text{ is a reducible constant term} \\ red(x) &= \begin{cases} \mu z\,(\langle u^- \rangle = z) & \text{if } \langle u \rangle = x \text{ and } R(x) \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

In particular, *red* is primitive recursive because we can compute all the reductions of u directly, and take the one with the smallest code. Moreover, $red(\langle u \rangle)$ is unique by definition, and if u is reducible then $red(\langle u \rangle) = \langle u^- \rangle$.

The whole point, given x coding a constant normal term, is to compute in a primitive recursive way the number of steps needed to reduce the term to a proper term, i.e.

$$\beta(x) = \mu z\,(red^{(z)}(x) \text{ is proper}),$$

where $red^{(z)}$ is the z-th iteration of *red*. After this we have the result, since to compute $f(\vec{x}, n)$ we consider the equivalent normal term $f(\vec{x}, p(n, n+1))$ and have

$$f(\vec{x}, n) = ev(red^{(\beta(y))}(\langle f(\vec{x}, p(n, n+1)) \rangle)),$$

which translates the fact that $f(\vec{x}, n)$ is the numerical value of the unique proper term obtained from $f(\vec{x}, p(n, n+1))$ by successive reductions.

We now concentrate on the definition of β. We would simply like to let

$$\beta(x) = \begin{cases} 0 & \text{if } \neg R(x) \\ 1 + \beta(red(x)) & \text{otherwise.} \end{cases}$$

The trouble is that it is not immediate that this is a primitive recursive function, since we have no reason to believe that $red(x) < x$. For example, the reduction of $f(\vec{x}, 1)$ is $h(\vec{x}, 0)$, which in principle might be an arbitrarily complicated term.

What we can do is to assign finite ordinals (natural numbers) $ord(x)$ to codes of normal terms x, in such a way that *ord* is primitive recursive, and

$$R(x) \Rightarrow ord(red(x)) < ord(x),$$

i.e. the ordinals are assigned in such a way that they decrease when a reducible term is reduced. Using this we will be able to perform a primitive recursion, if not directly on the codes of terms, at least on the finite ordinals assigned to them. In other words, we can define $\gamma(x, y)$ (where y should be thought of as $ord(x)$), as

$$\begin{aligned} \gamma(x, 0) &= 0 \\ \gamma(x, y+1) &= \begin{cases} 0 & \text{if } \neg R(x) \\ 1 + \gamma(red(x), p(ord(red(x)), y+1)) & \text{otherwise.} \end{cases} \end{aligned}$$

$\gamma(x, y{+}1)$ only uses a value $\gamma(red(x), z)$ with $z \le y$, due to the use of p, and this is a recursion with substitution of the function *red* in place of the parameter x, hence a primitive recursion by VIII.8.3.a. If we let

$$\beta(x) = \gamma(x, ord(x)),$$

then β is primitive recursive and

$$\beta(x) = \begin{cases} 0 & \text{if } \neg R(x) \\ 1 + \beta(red(x)) & \text{otherwise.} \end{cases}$$

Indeed, if $R(x)$ holds then $ord(red(x)) < ord(x)$, and so if $y + 1 = ord(x)$ we have $ord(red(x)) \le y$, hence $p(ord(red(x)), y + 1) = ord(red(x))$, and

$$\beta(x) = \gamma(x, ord(x)) = 1 + \gamma(red(x), ord(red(x))) = 1 + \beta(red(x)).$$

It only remains to assign the ordinals $ord(x)$. Let r be the number of all f-terms occurring in $h(\vec{x}, y)$.

- if $x = \langle u \rangle$ and u is not normal, let

$$ord(x) = 0$$

- if $x = \langle u \rangle$ and u is normal, consider the occurrences $f_1, \ldots, f_s$ of f-terms in it and, if $m_1, \ldots, m_s$ are the ordinals assigned to them, let

$$ord(x) = m_1 + \cdots + m_s$$

- if $x = \langle f(\vec{t}, p(s, n{+}1)) \rangle$, i.e. x codes a normal f-term, we define a primitive recursive function δ and let

$$ord(x) = \begin{cases} \delta(n+1) & \text{if } s \text{ is not proper} \\ r \cdot \delta(ev(\langle p(s, n+1) \rangle)) & \text{otherwise.} \end{cases}$$

  The reason is the following. In the first case we cannot evaluate the term directly, so we just indicate that our term may be used in the computation of $f(\vec{x}, n + 1)$. In the second case we can instead eliminate our term, according to the reduction rules, in favor of the r new occurrences of f-terms, each of them used to compute a value of the form $f(\vec{t}_j, ev(\langle p(s_j, n + 1) \rangle))$.

We still have to define δ in such a way that

$$R(x) \;\Rightarrow\; ord(red(x)) < ord(x).$$

Let u be a reducible term and let $f_1, \ldots, f_s$ be the f-terms occurring in it. The process of reduction consists of changing one f-term according to the rules, and this changes its ordinal. Since there might be nestings, this also produces possible changes in the ordinals of all the terms f_i containing the term which has been reduced. Since $ord(\langle u \rangle)$ and $ord(\langle u^- \rangle)$ are the sums of the ordinals associated with the f-terms occurring in u and u^-, it is enough to show that the reduction does not increase, and in at least one case it decreases, the ordinals of the individual f-terms f_i. There are two cases:

1. *f_i contains the term which is reduced*
   Let $f_i = f(\vec{t}, p(s, n+1))$. There are two cases:

   - *s is proper*
     Then nothing can change in s (i.e. the changes are in $\vec{t}$), and the ordinal associated with f_i is the same before and after the reduction, namely $r \cdot \delta(ev(\langle p(s, n+1) \rangle))$.
   - *s is not proper*
     If s is still not proper after the reduction, again the ordinal associated with f_i is the same before and after the reduction, namely $\delta(n+1)$.

     Otherwise, s becomes proper after the reduction, and the ordinal of f_i changes from $\delta(n+1)$ to $r \cdot \delta(ev(\langle p(s, n+1) \rangle))$. But
     $$ev(\langle p(s, n+1) \rangle) \leq n,$$
     and if δ is monotone, then
     $$\delta(ev(\langle p(s, n+1) \rangle)) \leq \delta(n).$$
     If we define δ in such a way that
     $$r \cdot \delta(n) \leq \delta(n+1),$$
     then
     $$r \cdot \delta(ev(\langle p(s, n+1) \rangle)) \leq \delta(n+1).$$

2. *f_i is the term which is reduced*
   Let $f_i = f(\vec{t}, p(s, n+1))$ where, by definition of reducibility, $\vec{t}$ and s are proper. There are two cases:

   - $ev(\langle p(s, n+1) \rangle) = 0$
     Then the term f_i is eliminated in favor of $g(\vec{t})$, and f no longer appears in it. The ordinal was $r \cdot \delta(0)$ before the reduction, and becomes 0 afterwards, so we want $\delta(0) > 0$.

- $ev(\langle p(s, n+1)\rangle) = n+1$
  We introduce $h(\vec{t}, n)$ in place of f_i, and have r new occurrences of f, of the form $f(\vec{t_j}, p(s_j, n+1))$. Each s_j may or may not be proper, but at least one (corresponding to an innermost occurrence of f) is proper. We thus have at most $r-1$ not proper f-terms, and at most r proper ones. By definition, we assign to $f(\vec{t_j}, p(s_j, n+1))$ the ordinal $\delta(n+1)$ if s_j is not proper, and $r \cdot \delta(ev(\langle p(s_j, n+1)\rangle))$ if s_j is proper. If, as above, δ is monotone, then $\delta(ev(\langle p(s_j, n+1)\rangle)) \leq \delta(n)$. Thus the ordinal assigned to $h(\vec{t}, n)$ is not greater than
  $$(r-1) \cdot \delta(n+1) + r^2 \cdot \delta(n).$$
  Also, the ordinal assigned to f_i is $r \cdot \delta(n+1)$, since s is proper and $ev(\langle p(s, n+1)\rangle) = n+1$, and thus we want
  $$(r-1) \cdot \delta(n+1) + r^2 \cdot \delta(n) < r \cdot \delta(n+1),$$
  i.e.
  $$r^2 \cdot \delta(n) < \delta(n+1).$$

The conditions required on δ are thus:

- δ is monotone
- $\delta(0) > 0$
- $r \cdot \delta(n) \leq \delta(n+1)$
- $r^2 \cdot \delta(n) < \delta(n+1)$.

It is then enough to define

$$\begin{aligned} \delta(0) &= 1 \\ \delta(n+1) &= r^2 \cdot \delta(n) + 1 \end{aligned}$$

to satisfy them. $\square$

The previous result can be expressed in the following form (reminiscent of the principle of *bar induction* in intuitionistic logic): primitive recursion on well-founded relations (of height ω) is reducible to primitive recursion on the usual well-ordering (of ordinal ω) of the natural numbers

A consequence of VIII.8.5 is that *the primitive recursive functions are exactly the functions that can be defined by recursion on ω on a single variable.* That there are recursive but not primitive recursive functions does not say that in the definition of primitive recursive functions we overlooked some kind of recursion on ω, but that besides recursions on ω there are other more general (abstract) kinds of recursions, that still produce computable functions. We will study some of them in Section 9.

A simple example of a more general recursion that leads out of $\mathcal{P}_1$ is nested recursion on *many* (actually on two) variables, as we see in VIII.8.11. On the other hand, *unnested* recursion on many variables does not lead out of $\mathcal{P}_1$, by VIII.8.3.b.

Alternative characterizations

We have already obtained two alternative characterizations of the class of primitive recursive functions in I.5.7 and I.5.10. Since the latter will be useful in the following, for the reader's convenience we repeat the relevant definition and proof.

Definition VIII.8.6 *A function f is defined by* **iteration** *from a function t if*

$$f(x,n) = t^{(n)}(x),$$

where $t^{(n)}(x)$ denotes the result of n successive applications of t (by convention, $t^{(0)}(x) = x$).

Proposition VIII.8.7 (Robinson [1947], Bernays) *The class of primitive recursive functions is the smallest class of functions:*

1. *containing the initial functions, together with coding and decoding functions for pairs*
2. *closed under composition*
3. *closed under iteration.*

Proof. Let $\mathcal{C}$ be the smallest class of functions satisfying the stated conditions. Then every function in $\mathcal{C}$ is primitive recursive, because the coding and decoding functions for pairs are primitive recursive, and the class of primitive recursive functions is closed under composition and iteration. Indeed, the latter is a special case of primitive recursion:

$$\begin{aligned} f(x,0) &= x \\ f(x,n+1) &= t(f(x,n)). \end{aligned}$$

For the converse, we only have to show that $\mathcal{C}$ is closed under primitive recursion. First, note that since $\mathcal{C}$ has coding and decoding functions for pairs and is closed under composition, it also has coding and decoding functions for n-tuples, for any fixed n. We continue to use the standard notations for coding and decoding functions. Let g and h be functions in $\mathcal{C}$, and

$$\begin{aligned} f(\vec{x},0) &= g(\vec{x}) \\ f(\vec{x},y+1) &= h(\vec{x},y,f(\vec{x},y)). \end{aligned}$$

We show that the function

$$s(\vec{x},n) = \langle \vec{x},n,f(\vec{x},n)\rangle$$

is in $\mathcal{C}$. Then so is f by composition, because

$$f(\vec{x},n) = (s(\vec{x},n))_{m+2}$$

(where m is the number of elements in the vector $\vec{x}$), and the decoding function is in $\mathcal{C}$.

Note that the function

$$\begin{aligned} s(\vec{x},0) &= \langle \vec{x},0,f(\vec{x},0)\rangle \\ &= \langle \vec{x},0,g(\vec{x})\rangle \end{aligned}$$

is in $\mathcal{C}$ by composition, since g and the coding function are in $\mathcal{C}$. Moreover,

$$\begin{aligned} s(\vec{x},n+1) &= \langle \vec{x},n+1,f(\vec{x},n+1)\rangle \\ &= \langle \vec{x},n+1,h(\vec{x},n,f(\vec{x},n))\rangle. \end{aligned}$$

Since

$$s(\vec{x},n) = \langle \vec{x},n,f(\vec{x},n)\rangle,$$

$s(\vec{x},n+1)$ can be obtained from $s(\vec{x},n)$ by one application of the function

$$t(\langle \vec{x},n,z\rangle) = \langle \vec{x},n+1,h(\vec{x},n,z)\rangle,$$

which is in $\mathcal{C}$ by composition, since g, h, the successor and the coding function are in $\mathcal{C}$. Finally,

$$s(\vec{x},n) = t^{(n)}(s(\vec{x},0)),$$

and $s(\vec{x},n)$ is thus the composition of $s(\vec{x},0)$ and the iteration of t. Since these are both in $\mathcal{C}$, so is s. $\square$

A number of further improvements and refinements have been considered on pp. I.73–74.

Computation time and space

From a computational point of view the properties of $\mathcal{P}_1$ are similar to those of $\mathcal{E}$, and the next result is the analogue of VIII.7.6.

Theorem VIII.8.8 Ritchie-Cobham Property for $\mathcal{P}_1$ (Kleene [1936], Cobham [1964], Meyer [1965]) *The following are equivalent:*

1. *f is primitive recursive*

2. *f is computable in primitive recursive time*

3. *f is computable in primitive recursive space.*

Proof. Parts 2 and 3 are equivalent by VII.4.17, since $\mathcal{P}_1$ is closed under exponential (by VIII.8.2). It is thus enough to prove the equivalence between Parts 1 and 2.

If f is primitive recursive, then the first part of the proof of VIII.7.6 shows that it is computable in primitive recursive time. More precisely, it shows that if f is defined by primitive recursion from g and h, i.e.

$$\begin{aligned} f(\vec{x},0) &= g(\vec{x}) \\ f(\vec{x},z+1) &= h(\vec{x},z,f(\vec{x},z)), \end{aligned}$$

and T_g and T_h are the times needed to compute g and h, then the time T_f needed to compute f is roughly

$$T_f(\vec{x},z) = T_g(\vec{x}) + \sum_{y<z} T_h(\vec{x},y,f(\vec{x},y)).$$

If f, T_g and T_h are primitive recursive, then so is T_f, by VIII.8.2.

Conversely, the second part of the proof of VIII.7.6 shows that every recursive function is elementary in the time needed to compute it. If f is computable in primitive recursive time t, then f is elementary in a primitive recursive function, and by VIII.8.2, f is primitive recursive. $\square$

Corollary VIII.8.9 (Meyer and McCreight [1969]) *$\mathcal{P}_1$ is a complexity class with respect to time, space and any measure primitive recursively related to them.*

Proof. From VIII.8.8, as in VIII.7.7. $\square$

Rate of growth

One possible strategy for obtaining a hierarchy for the primitive recursive functions comes from the approach used for the elementary functions (VIII.7.8), and requires a sequence of primitive recursive functions that is cofinal in the class of primitive recursive functions.

One possible tactic to define such a sequence comes from the characterization of the class of primitive recursive functions in terms of iteration (VIII.8.7). Since the initial functions are all dominated by the successor function, iterations of it should dominate all functions defined by iterations from the initial functions. The next result confirms this guess.

Theorem VIII.8.10 A Skeleton for $\mathcal{P}_1$ (Sudan [1927], Ackermann [1928], Peter [1935], Robinson [1948]) *Let*

$$h_0(x) = x + 1 \qquad \textit{and} \qquad h_{n+1}(x) = h_n^{(x)}(x).$$

Then:

1. *h_n is primitive recursive, for each n*
2. *every primitive recursive function is dominated by some h_n, in the sense that given f primitive recursive, there is an n such that $f(\vec{x}) \leq h_n(\sum \vec{x})$, for almost every $\vec{x}$*
3. *the diagonal function $d_{\mathcal{P}_1}(x) = h_x(x)$ dominates every primitive recursive function, and thus it is not primitive recursive.*

Proof. Before turning to the proof itself, we notice the following properties of the functions h_n:

1. $x \leq h_n(x)$
   By induction on n. For $n = 0$, this is trivial. For $n+1$, suppose $z \leq h_n(z)$ for every z. Then $x \leq h_n^{(m)}(x)$ follows trivially by induction on m, and in particular
   $$x \leq h_n^{(x)}(x) = h_{n+1}(x).$$
2. *if $n \leq m$ and $x > 0$, then $h_n(x) \leq h_m(x)$*
   It is enough to show that $h_n(x) \leq h_{n+1}(x) = h_n^{(x)}(x)$, for $x > 0$. For this it is enough to show that $h_n(z) \leq h_n^{(m)}(z)$, by induction on $m > 0$. For $m = 1$, this is trivial. For $m + 1$,
   $$\begin{array}{rcll} h_n(z) & \leq & h_n^{(m)}(z) & \text{by induction hypothesis} \\ & \leq & h_n(h_n^{(m)}(z)) & \text{by Property 1 above} \\ & = & h_n^{(m+1)}(z). & \end{array}$$

3. *if $x \leq y$, then $h_n(x) \leq h_n(y)$*
   By induction on n. For $n = 0$ this is trivial. For $n + 1$,

$$\begin{array}{rcll} h_{n+1}(x) & = & h_n^{(x)}(x) & \\ & \leq & h_n^{(y)}(x) & \text{by Property 1 above } (y - x \text{ times}) \\ & \leq & h_n^{(y)}(y) & \text{by induction hypothesis } (y \text{ times}) \\ & = & h_{n+1}(y). & \end{array}$$

4. *$h_1(x) = 2x$ and $h_2(x) = x \cdot 2^x$*
   To prove $h_1(x) = 2x$ it is enough to show that $h_0^{(m)}(x) = x + m$, by induction on m. For $m = 0$ this is trivial. For $m + 1$,

$$\begin{array}{rcll} h_0^{(m+1)}(x) & = & h_0(h_0^{(m)}(x)) & \\ & = & h_0(x + m) & \text{by induction hypothesis} \\ & = & (x + m) + 1 & \text{since } h_0(z) = z + 1 \\ & = & x + (m + 1). & \end{array}$$

   To prove $h_2(x) = x \cdot 2^x$ it is enough to show that $h_1^{(m)}(x) = x \cdot 2^m$, by induction on m. For $m = 0$ this is trivial. For $m + 1$,

$$\begin{array}{rcll} h_1^{(m+1)}(x) & = & h_1(h_1^{(m)}(x)) & \\ & = & h_1(x \cdot 2^m) & \text{by induction hypothesis} \\ & = & (x \cdot 2^m) \cdot 2 & \text{since } h_1(z) = 2z \\ & = & x \cdot 2^{m+1}. & \end{array}$$

We now turn to the proof.

Part 1 is obvious, since the successor function is primitive recursive, and $\mathcal{P}_1$ is closed under iteration.

Part 3 follows from Part 2 and Property 2 above, since $h_n(x) \leq h_x(x)$ for every n and $x \geq n$.

Part 2 is proved by induction on the characterization of primitive recursive functions given by VIII.8.7, as follows:

- *initial functions, coding and decoding functions*
  They are all elementary, and thus dominated by iterations of the exponential function 2^x by VIII.7.8. But $h_2(x) = x \cdot 2^x$ by Property 4 above, and thus all elementary functions are dominated by iterations of h_2, and hence by h_3.

- *composition*
  If f has been defined by composition from $g_1, \ldots, g_m$ and h, i.e.

$$f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x})),$$

by induction hypothesis there is an $n \geq 2$ such that $g_i(\vec{x}) \leq h_n(\sum \vec{x})$ for almost every $\vec{x}$, and $h(\vec{y}) \leq h_n(\sum \vec{y})$ for almost every $\vec{y}$ (actually, the induction hypothesis gives different n's for $g_1, \ldots, g_m$ and h, but Property 2 above shows that their maximum works for all of them simultaneously). Then

$$\begin{array}{rcll} f(\vec{x}) & = & h(g_1(\vec{x}), \ldots, g_m(\vec{x})) & \\ & \leq & h_n(\sum_{1 \leq i \leq m} g_i(\vec{x})) & \text{by induction hypothesis on } h \\ & \leq & h_n(\sum_{1 \leq i \leq m} h_n(\sum \vec{x})) & \text{by induction hypothesis on } g_i \\ & & & \text{and Property 3} \\ & \leq & h_n(m \cdot h_n(\sum \vec{x})) & \\ & \leq & h_n(h_2(h_n(\sum \vec{x}))) & \text{if } m \leq \sum \vec{x}, \text{ since } h_2(z) = z \cdot 2^z \\ & \leq & h_n(h_n(h_n(\sum \vec{x}))) & \text{by Properties 2 and 3, since } n \geq 2 \\ & = & h_n^{(3)}(\sum \vec{x}) & \\ & \leq & h_n^{(\sum \vec{x})}(\sum \vec{x}) & \text{if } 3 \leq \sum \vec{x}, \text{ by Property 1} \\ & = & h_{n+1}(\sum \vec{x}). & \end{array}$$

- *iteration*
  If f has been defined by iteration from t, i.e.

$$f(x, y) = t^{(y)}(x),$$

  by induction hypothesis there is an n such that $t(x) \leq h_n(x)$ for almost every x. Then

$$\begin{array}{rcll} f(x, y) & = & t^{(y)}(x) & \\ & \leq & h_n^{(y)}(x) & \text{by induction hypothesis } (y \text{ times}) \\ & & & \text{and Property 3} \\ & \leq & h_n^{(x)}(x) & \text{if } y \leq x, \text{ by Property 1 } (x - y \text{ times}) \\ & = & h_{n+1}(x) & \\ & \leq & h_{n+1}(x + y) & \text{by Property 3.} \quad \Box \end{array}$$

Corollary VIII.8.11 *$\mathcal{P}_1$ is not closed under nested recursion on two variables.*

Proof. The sequence $\{h_n\}_{n \in \omega}$ of VIII.8.10 can be described by the following equations, the first two defining h_n by recursion on n, and the second two defining $h_n^{(z)}$ by recursion on z:

$$\begin{array}{rcl} h_0(x) & = & x + 1 \\ h_{n+1}(x) & = & h_n^{(x)}(x) \\ h_n^{(0)}(x) & = & x \\ h_n^{(z+1)}(x) & = & h_n(h_n^{(z)}(x)). \end{array}$$

They can be seen as defining a single function

$$h(n, z, x) = h_n^{(z)}(x),$$

by nested recursion on the two variables n and x, as follows (thinking of h_n as $h_n^{(1)}$):

$$\begin{array}{rcl} h(n, 0, x) & = & x \\ h(0, 1, x) & = & x + 1 \\ h(n+1, 1, x) & = & h(n, x, x) \\ h(n, z+1, x) & = & h(n, 1, h(n, z, x)). \end{array}$$

If h were primitive recursive then so would be the diagonal function

$$d_{\mathcal{P}_1}(x) = h_x(x) = h(x, 1, x),$$

and this would contradict VIII.8.10.3. So h is not primitive recursive, and $\mathcal{P}_1$ is not closed under nested recursion on two variables. □

Hierarchies

By using VIII.8.10, we can immediately obtain hierarchies for $\mathcal{P}_1$. Recall that $\mathcal{E}(g)$ is the class of functions elementary in g (VIII.7.16).

Definition VIII.8.12 The Grzegorczyk Hierarchy (Grzegorczyk [1953]) *For $n \geq 2$, $\boldsymbol{\mathcal{E}_{n+1}} = \mathcal{E}(h_n)$.*

The reason why we consider only $n \geq 2$ is that h_0, h_1 and h_2 are elementary, being respectively $x + 1$, $2x$ and $x \cdot 2^x$, and thus

$$\mathcal{E}(h_0) = \mathcal{E}(h_1) = \mathcal{E}(h_2) = \mathcal{E}_3 = \mathcal{E}.$$

The reason why we do not define $\mathcal{E}_n = \mathcal{E}(h_{n+2})$ for every n is purely historical: Grzegorczyk [1953] had a different definition of the classes $\mathcal{E}_n$, equivalent to that given above for $n \geq 3$, and we do not want to change a well-established notation. The original classes $\mathcal{E}_0$, $\mathcal{E}_1$ and $\mathcal{E}_2$ are defined in VIII.8.20.

Theorem VIII.8.13 Hierarchy Theorem for $\boldsymbol{\mathcal{P}_1}$ (Grzegorczyk [1953])

1. *$\mathcal{E}_n \subset \mathcal{E}_{n+1}$, for each $n \geq 3$*
2. *$\mathcal{P}_1 = \bigcup_{n \geq 3} \mathcal{E}_n$.*

Proof. Part 1 follows from the following two facts, for $n \geq 2$:

- *h_n is elementary in h_{n+1} (so $\mathcal{E}_{n+1} \subseteq \mathcal{E}_{n+2}$)*
  We prove by induction on m that if $m \leq n$, then h_m is elementary in h_n.

  For $m = 0$, h_0 is elementary, and hence elementary in any function.

  For $m+1$, if $m+1 \leq n$, then h_{m+1} is bounded by h_n by Property 2 in VIII.8.10, h_m is elementary in h_n by induction hypothesis, and h_{m+1} is elementary in h_n because it is obtained from h_m by (iteration, and hence by) primitive recursion bounded by h_n.

- *h_{n+1} is not elementary in h_n (so $\mathcal{E}_{n+1} \neq \mathcal{E}_{n+2}$)*
  For $n \geq 2$, h_{n+1} is not elementary in h_n by relativization of VIII.7.8 (see VIII.7.21.b), because $h_n(x) \geq 2^x$ (by Property 2 in the proof of VIII.8.10, since $n \geq 2$) and h_{n+1} is the iteration of h_n.

To prove Part 2, first note that $\bigcup_{n \geq 3} \mathcal{E}_n \subseteq \mathcal{P}_1$ because $\mathcal{P}_1$ is closed under elementary operations by VIII.8.2, and each h_n is primitive recursive by VIII.8.10.1. Conversely, $\mathcal{P}_1 \subseteq \bigcup_{n \geq 3} \mathcal{E}_n$ follows by induction on the definition of $\mathcal{P}_1$:

- *initial functions*
  They are all elementary, and hence in $\mathcal{E}_3$.

- *composition*
  If f has been defined by composition from $g_1, \ldots, g_m$ and h, by induction hypothesis there is an $n \geq 3$ such that g_i and h are all in $\mathcal{E}_n$ (the induction hypothesis gives different n's for $g_1, \ldots, g_m$ and h, but the inclusion property among the classes $\mathcal{E}_n$ proved in Part 1 shows that their maximum works for all of them simultaneously). Then also f is in $\mathcal{E}_n$, because by definition $\mathcal{E}_n$ is closed under (elementary operations, and hence under) composition.

- *primitive recursion*
  If f has been defined by primitive recursion from g and h, by induction hypothesis there is an $n \geq 2$ such that g and h are in $\mathcal{E}_{n+1}$. Since f is primitive recursive, f is dominated by h_m for some m, by VIII.8.10.2, and in the proof of VIII.8.10 we may suppose $m \geq n$, by Property 2. Then f is dominated everywhere by a finite modification of h_m, hence by a function in $\mathcal{E}_{m+1}$, and it is thus defined by bounded primitive recursion from functions in $\mathcal{E}_{m+1}$. Then also f is in $\mathcal{E}_{m+1}$, because by definition $\mathcal{E}_{m+1}$ is closed under (elementary operations, and hence under) bounded primitive recursion. $\square$

It follows from the Hierarchy Theorem for $\mathcal{P}_1$ that *there is no primitive recursive (in particular, no finite) set of primitive recursive functions that generates $\mathcal{P}_1$ by composition alone*, since the classes $\mathcal{E}_n$ are closed under composition, and any primitive recursive set of primitive recursive functions is contained in $\mathcal{E}_n$ for some fixed n.

From a computational point of view the properties of the classes of the Grzegorczyk Hierarchy are similar to those of $\mathcal{E}$ and $\mathcal{P}_1$, and the next result is the analogue of VIII.7.6 and VIII.8.8.

Theorem VIII.8.14 Ritchie-Cobham Property for $\mathcal{E}_n$ (Cobham [1964], Meyer [1965]) *For any $n \geq 3$, the following are equivalent:*

1. *f belongs to $\mathcal{E}_n$*
2. *f is computable in time belonging to $\mathcal{E}_n$*
3. *f is computable in space belonging to $\mathcal{E}_n$.*

Proof. Since, for $n \geq 2$, $\mathcal{E}_{n+1}$ is the class of functions elementary in h_n, by VIII.7.18 it is enough to prove, by induction on n, the following:

- *h_n is computable in time elementary in h_n*
  For $n = 0$, h_0 is elementary by definition, and computable in elementary time by VIII.7.6.

  For $n+1$, since $h_{n+1}(x) = h_n^{(x)}(x)$ it is enough to prove that the function $h(n, x, z) = h_n^{(z)}(x)$ is computable in time elementary in h_{n+1}, for each fixed n. By definition,

$$\begin{aligned} h(n, x, 0) &= x \\ h(n, x, z+1) &= h_n(h(n, x, z)). \end{aligned}$$

  Exactly as in the proof of VIII.7.6, if T_{h_n} and T_h are the times needed to compute h_n and h and we suppose that no time is needed to compute the identity function (used in the definition), then roughly

$$T_h(n, x, z) = \sum_{y<z} T_{h_n}(h(n, x, y)).$$

  The properties of h_n shown in the proof of VIII.8.10 imply that

$$h(n, x, y) = h_n^{(y)}(x) \leq h_n^{(x+y)}(x+y) = h_{n+1}(x+y),$$

  so

$$T_{h_n}(h(n, x, y)) \leq \max\{T_{h_n}(s) : s \leq h_{n+1}(x+y)\}$$

  and

$$T_h(n, x, z) \leq \sum_{y<z} \max\{T_{h_n}(s) : s \leq h_{n+1}(x+y)\}.$$

By induction hypothesis, h_n is computable in time elementary in h_n, i.e. T_{h_n} is elementary in h_n, and hence T_h is bounded by a function elementary in h_{n+1}, for each fixed n. $\Box$

Corollary VIII.8.15 (Meyer and McCreight [1969]) *For any $n \geq 3$, $\mathcal{E}_n$ is a complexity class with respect to time and space.*

Proof. From VIII.8.14, as in VIII.7.7. $\Box$

Exercises VIII.8.16 a) *For any $n \geq 3$, $\mathcal{E}_{n+1}$ contains a universal function for $\mathcal{E}_n$.* (Grzegorczyk [1953]) (Hint: since $n \geq 3$, the coding apparatus is in $\mathcal{E}_{n+1}$. Using the fact that h_n, which is in $\mathcal{E}_{n+1}$, dominates every function of $\mathcal{E}_n$, we can build a universal function for $\mathcal{E}_n$ in $\mathcal{E}_{n+1}$ by arithmetization and bounded μ-recursion, as in I.7.4.)

b) *For any $n \geq 3$, the canonical universal function for $\mathcal{E}_n$ is elementarily equivalent to h_n.* (Axt [1959]) (Hint: one direction comes from part a). Conversely, if the enumeration $\{f_e\}_{e\in\omega}$ is canonical then there is an elementary function g such that $f_{g(z)}(x) = h_{n-1}^{(z)}(x)$. But then $h_n(x) = f_{g(x)}(x)$, and h_n is elementary in $\{f_e\}_{e\in\omega}$.)

c) *For any $n \geq 3$, $\mathcal{E}_{n+1} - \mathcal{E}_n$ contains a 0,1-valued function.* (Grzegorczyk [1953]) (Hint: let $\{f_e\}_{e\in\omega}$ be an enumeration of $\mathcal{E}_n$ in $\mathcal{E}_{n+1}$. Then the diagonal function $g(x) = 1 - f_x(x)$ is in $\mathcal{E}_{n+1}$ by trivial closure properties, but not in $\mathcal{E}_n$ by diagonalization. Alternative proofs can be given using the Space or Time Hierarchy Theorems (VII.4.13 and VII.4.15), together with VIII.8.14).

Meyer and Ritchie [1972] show that for elementarily honest classes there is a natural connection among dominating functions, enumerating functions, set-theoretical containment and the existence of 0,1-valued functions in the difference, of which the exercises above are examples (for the classes $\mathcal{E}_n$).

By VIII.8.14, the Grzegorczyk Hierarchy classifies the primitive recursive functions according to their computation time or space, as measured w.r.t. the sequence $\{h_n\}_{n\in\omega}$.

Another natural classification is based on the number of primitive recursions needed to define a primitive recursive function.

Definition VIII.8.17 The Primitive Recursion Hierarchy (Heinermann [1961], Axt [1965]) *For $n \geq 2$, the class $\boldsymbol{\mathcal{A}_n}$ is defined by induction, as follows:*

1. *$\mathcal{A}_2 = \mathcal{E}$*
2. *$\mathcal{A}_{n+1}$ is the smallest class of functions containing $\mathcal{A}_n$, and closed under composition and a single application of the primitive recursion schema.*

Once again we use $n \geq 2$ only for historical reasons, and the original classes $\mathcal{A}_0$ and $\mathcal{A}_1$ are defined in VIII.8.20.

Exercises VIII.8.18 a) $\mathcal{P}_1 = \bigcup_{n\geq 2} \mathcal{A}_n$.

b) *For any $n \geq 2$, $\mathcal{A}_n \subset \mathcal{A}_{n+1}$.* (Axt [1965], [1966]) (Hint: if $\mathcal{A}_n = \mathcal{A}_{n+1}$ for some $n \geq 2$, then $\mathcal{P}_1 = \mathcal{A}_n$. But a primitive recursive function not in $\mathcal{A}_n$ exists, by a domination argument.)

It is quite interesting that the two hierarchies for $\mathcal{P}_1$ introduced above, although defined in quite different ways, coincide.

Proposition VIII.8.19 (Axt [1965]) *For any $n \geq 2$, $\mathcal{A}_n = \mathcal{E}_{n+1}$.*

Proof. We have to prove inclusion in both directions.

- *for any $n \geq 2$, $\mathcal{A}_n \subseteq \mathcal{E}_{n+1}$*
  By induction on $n \geq 2$. For $n = 2$, this holds by definition. For $n + 1$, suppose $\mathcal{A}_n \subseteq \mathcal{E}_{n+1}$. To prove $\mathcal{A}_{n+1} \subseteq \mathcal{E}_{n+2}$, it is enough to show that every function defined by a single primitive recursion from functions in $\mathcal{E}_{n+1}$ belongs to $\mathcal{E}_{n+2}$. By the proof of VIII.8.7, every application of the primitive recursion schema is reducible to an application of the iteration schema modulo elementary functions (needed for coding and decoding purposes). We then have to prove that if the function t used in the iteration is in $\mathcal{E}_{n+1}$, then its iteration is in $\mathcal{E}_{n+2}$.

  To see this, it is enough to show that the iteration of t is bounded by a function in $\mathcal{E}_{n+2}$, since then it is defined by bounded primitive recursion in $\mathcal{E}_{n+2}$. But if t is in $\mathcal{E}_{n+1} = \mathcal{E}(h_n)$, then by relativization of VIII.7.8 (see VIII.7.21.b) there is a finite iteration of h_n dominating t, i.e.

  $$t(x) \leq h_n^{(m)}(x)$$

  for some m and almost every x. Thus, by the properties of h_n in the proof of VIII.8.10,

  $$t^{(z)}(x) \leq h_n^{(m+z)}(x) \leq h_n^{(m+x+z)}(m + x + z) = h_{n+1}(m + x + z)$$

  for some m, every z and almost every x. Thus the iteration of t is dominated everywhere by a finite modification of h_{n+1}, hence by a function elementary in h_{n+1}, which is in $\mathcal{E}_{n+2}$.

- *for any $n \geq 2$, $\mathcal{E}_{n+1} \subseteq \mathcal{A}_n$*
  This follows immediately by induction on n, since $\mathcal{E}_3 = \mathcal{A}_2$, and each h_{n+1} is obtained from h_n by a single application of the primitive recursion schema. □

Exercises VIII.8.20 The lower levels of the Grzegorczyk and Primitive Recursion Hierarchies. The class $\mathcal{E}_0$ is defined as the smallest class of functions containing the initial functions, and closed under composition and bounded recursion. The classes $\mathcal{E}_1$ and $\mathcal{E}_2$ are defined similarly, by respectively adding $x+y$ and $x \cdot y$ to the initial functions.

The class $\mathcal{A}_0$ is defined as the smallest class of functions containing the initial functions, and closed under composition. The classes $\mathcal{A}_1$ and $\mathcal{A}_2$ are obtained inductively, as in VIII.8.17.

a) $\mathcal{E}_0 \subset \mathcal{E}_1 \subset \mathcal{E}_2 \subset \mathcal{E}$. (Grzegorczyk [1953]) (Hint: every function $f(x)$ in $\mathcal{E}_0$ is bounded by $x+c$ for some constant c, and hence $2x$ is not in $\mathcal{E}_0$. Similarly, x^2 is not in $\mathcal{E}_1$, and 2^x is not in $\mathcal{E}_2$. Also, $\mathcal{E}_1 \subseteq \mathcal{E}_2$ because $x+y$ can be defined by primitive recursion bounded by $(x+1)\cdot(y+1)$, from the successor function.)

b) $\mathcal{A}_0 \subset \mathcal{A}_1 \subset \mathcal{A}_2 \subseteq \mathcal{E}$. (Axt [1965]) (Hint: every function $f(x)$ in $\mathcal{A}_0$ is bounded by $x+c$ for some constant c, and hence $2x$ is not in $\mathcal{A}_0$. Every function $f(x)$ in $\mathcal{A}_1$ is bounded by $c \cdot x + d$ for some constants c and d, and hence 2^x is not in $\mathcal{A}_1$. $\mathcal{A}_2 \subseteq \mathcal{E}$ because every function in $\mathcal{A}_2$ is dominated by a finite number of iterations of the exponential function.)

c) $\mathcal{A}_0 \subset \mathcal{E}_0$. (Axt [1965]) (Hint: if $f(x)$ is in $\mathcal{A}_0$, then there is a constant c such that $f(x) = c$, or $f(x) = x + c$, and thus the predecessor function is in $\mathcal{E}_0$ but not in $\mathcal{A}_0$.)

d) $\mathcal{A}_1$ *and* $\mathcal{E}_0$ *are incomparable, and* $\mathcal{A}_1 \subset \mathcal{E}_1$. (Axt [1965]) (Hint: if $f(x)$ is in $\mathcal{A}_1$, then there are constants c and d such that $f(x) = c \cdot x + d$, and hence $\frac{x}{2}$ is in $\mathcal{E}_0$ but not in $\mathcal{A}_1$, so that $\mathcal{E}_0 \not\subseteq \mathcal{A}_1$ and $\mathcal{A}_1 \subset \mathcal{E}_1$. Moreover, $x+y$ is in $\mathcal{A}_1$ but not in $\mathcal{E}_0$, so that $\mathcal{A}_1 \not\subseteq \mathcal{E}_0$.)

A series of results by Meyer [1965], D. Ritchie [1965], Schwichtenberg [1969], and Muller [1973] has led to a proof that $\mathcal{A}_2 = \mathcal{E}$, which justifies the definition in VIII.8.17. In particular, every elementary function can be obtained from the initial functions by composition and only two applications of the primitive recursion schema. Note that 2^x can be obtained by two primitive recursions, the first defining addition and the second defining $2^0 = 1$ and $2^{x+1} = 2^x + 2^x$. By composition, also $\exp_n$ (defined in VIII.7.8) is in $\mathcal{A}_2$, for every n.

We thus obtain the following picture, where the only inclusions are those indicated:

$$\mathcal{A}_0 \subset \begin{array}{c} \mathcal{A}_1 \\ \\ \mathcal{E}_0 \end{array} \subset \mathcal{E}_1 \subset \mathcal{E}_2 \subset \mathcal{A}_2 = \mathcal{E}_3 = \mathcal{E}.$$

For more information on the properties of the lower classes of the various hierarchies and their relationships, see Grzegorczyk [1953], Skolem [1954a], [1962], [1963], Moh [1956], Smullyan [1961], Bennett [1962], R. Ritchie [1963], [1965], Trakhtenbrot [1964], D. Ritchie [1965], Nepomniaschki [1966], [1966a], [1970], [1970a], Kozmiadi and Marchenkov [1969], Marchenkov [1969], [1975a], Tsichritzis [1970], [1971], Thompson [1972], Harrow [1975], Goetze and Nehrlich [1981], Gandy [1984]. In particular, an Improved Normal Form Theorem for partial recursive functions has been obtained, with $\mathcal{U}$ and $\mathcal{T}_n$ in $\mathcal{E}_0$. This result is presently incomparable with VIII.2.8, since no relationship is known to hold between $\mathcal{E}_0$ and PF (see also the following remarks).

Let $(\mathcal{E}_n)_*$ be the class of 0,1-valued functions in $\mathcal{E}_n$. Then

$$(\mathcal{E}_0)_* \subseteq (\mathcal{E}_1)_* \subseteq (\mathcal{E}_2)_* \subset (\mathcal{E}_3)_* \subset (\mathcal{E}_4)_* \subset \cdots$$

where the first proper inclusion has been proved by Ritchie [1963], and those following it have been proved in VIII.8.16.c. It is not known whether either of the first two inclusions is proper.

In terms of the standard complexity theoretic classes, Smullyan [1961] and Bennett [1962] have proved that $\Delta_0^0 \subseteq (\mathcal{E}_0)_*$, so that it follows from the result quoted on p. 225 that LOGSPACE $\subseteq (\mathcal{E}_0)_*$ (none of the two inclusions is known to be proper). At the opposite extreme, Ritchie [1963] has proved that $(\mathcal{E}_2)_* =$ LINSPACE, so that $(\mathcal{E}_2)_* \subset PSPACE$ (by VIII.5.5) and $(\mathcal{E}_2)_* \neq P$ (by VIII.5.9).

Exercises VIII.8.21 The Loop Hierarchy. (Meyer and Ritchie [1967]) We have seen in I.5.8 that the primitive recursive functions are exactly those computable by 'for' programs, and this provides a further way of classifying them. Namely, let

$$\mathcal{L}_n = \{\text{functions definable by 'for' programs with at most } n \text{ nestings of 'for'}\}.$$

a) *The simulation of a 'for' program can be done by functions in* $\mathcal{L}_2$. (Hint: the values of the variables at a given instant, as well as the number of the instruction applying, can be defined by simultaneous recursion, using case definitions (see p.$\tilde{\text{I}}$.71). This amounts to a definition using only two nestings of 'for', one being needed for the case definition, and another for the simultaneous recursion.)

b) $\mathcal{E} \subseteq \mathcal{L}_2$. (Hint: as in VIII.7.6 and by induction on the definition of $\mathcal{E}$, we can prove that if f is elementary, then it is computable by a 'for' program in elementary time (i.e. in an elementary number of moves) T_f. But T_f is then bounded by a finite iteration of h_2, hence by a function in $\mathcal{L}_2$ (see VIII.8.20 for how the exponential function can be defined by two primitive recursions). By part a), f is then in $\mathcal{L}_2$.)

c) *For any* $n \geq 2$, $\mathcal{L}_n = \mathcal{E}_{n+1}$. (Hint: simultaneous recursion is equivalent, modulo composition and elementary functions, to primitive recursion. Then from $\mathcal{A}_2 \subseteq \mathcal{E}$ (see VIII.8.20.b), we have $\mathcal{L}_2 \subseteq \mathcal{E}$ and, by part b), $\mathcal{L}_2 = \mathcal{E}$. By induction and VIII.8.19, $\mathcal{L}_n = \mathcal{E}_{n+1}$ follows for any $n \geq 2$.)

A thorough study of programs for primitive recursive functions has been done by Cleave [1963], Constable [1971], Constable and Borodin [1972], Constable and Muchnik [1972], Meyer [1972] and Tsichritzis [1970].

The results of the present subsection show that the Grzegorczyk Hierarchy has a strong stability, and it can be defined in various and conceptually different ways, which turn out to be equivalent. We have seen definitions based on rate of growth (VIII.8.12), computation time and space (VIII.8.14), enumeration (VIII.8.16.b), number of recursions (VIII.8.17), nesting of 'for' instructions (VIII.8.21). Thus, knowledge of membership of a function in a class $\mathcal{E}_n$ is very informative.

A number of other characterizations have been obtained, see Axt [1963], Ritchie [1965a], Cleave and Rose [1967], Parsons [1968], Kozmiadi [1970], Muchnik [1976], Harrow [1979], Facchini and Maggiolo-Schettini [1982], Goetze and Nehrlich [1980]. All these approaches coincide as soon as they reach the class of elementary functions, although they differ at their lower levels.

The Grzegorczyk Hierarchy can also be refined by building hierarchies inside $\mathcal{E}_n$ for $n \geq 3$, some of them modelled on the hierarchies studied in the previous section for $\mathcal{E}$. See VIII.8.24 and Cleave [1963], Kazanovich [1970], Goetze and Nehrlich [1978], [1980].

The diagonal function $\star$

In this subsection we analyze the rate of growth of the diagonal function $d_{\mathcal{P}_1}$, defined in VIII.8.10, as we did for $d_{\mathcal{E}}$ in VIII.7.14.

Before turning to the computation of the position of $d_{\mathcal{P}_1}$ in the Slow Growing Hierarchy defined in VIII.7.12, we briefly consider a sequence of faster-growing functions (already mentioned briefly in VIII.7.15), which will later turn out to be useful for comparison.

Definition VIII.8.22 The Moderate Growing Hierarchy (Hardy [1904], Wainer [1972]) *The sequence $\{g_\alpha\}_\alpha$ is defined by induction on the countable ordinals, as follows:*

$$\begin{array}{rcll} g_0(x) & = & x & \\ g_{\alpha+1}(x) & = & g_\alpha(x+1) & \\ g_\alpha(x) & = & g_{\alpha_x}(x) & \quad \textit{if } \alpha \textit{ is a limit} \end{array}$$

where, for α a limit, $\{\alpha_x\}_{x\in\omega}$ is an increasing (fundamental) sequence of ordinals with limit α.

In the previous definition, we use the fundamental sequences defined in VIII.7.13, when $\alpha \leq \epsilon_0$.

Notice that the crucial difference between the Slow and Moderate Growing Hierarchies lies in the successor case, where

$$f_{\alpha+1}(x) = f_\alpha(x) + 1 \qquad \text{and} \qquad g_{\alpha+1}(x) = g_\alpha(x+1).$$

In both cases we compose the function obtained at the previous level with the successor function, but in reverse order:

$$f_{\alpha+1} = \mathcal{S} \circ f_\alpha \qquad \text{and} \qquad g_{\alpha+1} = g \circ \mathcal{S}.$$

As we might expect, composing internally produces a better behavior, in particular it allows for a simple representation of composition of two g_α's, a result that does not hold for the f_α's.

The next result is an analogue of VIII.7.14.

Proposition VIII.8.23 The Position of $d_{\mathcal{P}_1}$ in the Moderate Growing Hierarchy (Wainer [1972])

1. $h_n = g_{\omega^n}$

2. $d_{\mathcal{P}_1} = g_{\omega^\omega}$.

Proof. The result follows from the following properties:

1. $g_{\alpha+\beta}(x) = g_\alpha(g_\beta(x))$, *for every* $\alpha, \beta < \epsilon_0$

   This is proved by induction on β:

   - $\beta = 0$

$$\begin{array}{rcll} g_{\alpha+0}(x) & = & g_\alpha(x) & \\ & = & g_\alpha(g_0(x)) & \text{by definition of } g_0. \end{array}$$

   - $\beta = \gamma + 1$

$$\begin{array}{rcll} g_{\alpha+(\gamma+1)}(x) & = & g_{(\alpha+\gamma)+1}(x) & \text{by associativity of } + \\ & = & g_{\alpha+\gamma}(x+1) & \text{by definition of } g_{(\alpha+\gamma)+1} \\ & = & g_\alpha(g_\gamma(x+1)) & \text{by induction hypothesis} \\ & = & g_\alpha(g_{\gamma+1}(x)) & \text{by definition of } g_{\gamma+1}. \end{array}$$

   - β *a limit*

$$\begin{array}{rcll} g_{\alpha+\beta}(x) & = & g_{(\alpha+\beta)_x}(x) & \text{by definition of } g_{\alpha+\beta} \\ & = & g_{\alpha+\beta_x}(x) & \text{by definition of } (\alpha+\beta)_x \\ & = & g_\alpha(g_{\beta_x}(x)) & \text{by induction hypothesis} \\ & = & g_\alpha(g_\beta(x)) & \text{by definition of } g_\beta. \end{array}$$

2. $g_{\omega^{\alpha+1}}(x) = g_{\omega^\alpha}^{(x)}(x)$, *for every* $\alpha < \epsilon_0$

$$\begin{array}{rcll} g_{\omega^{\alpha+1}}(x) & = & g_{(\omega^{\alpha+1})_x}(x) & \text{by definition of } g_{\omega^{\alpha+1}} \\ & = & g_{\omega^\alpha \cdot x}(x) & \text{by definition of } (\omega^{\alpha+1})_x \\ & = & g_{\underbrace{\omega^\alpha + \cdots + \omega^\alpha}_{x \text{ times}}}(x) & \text{by definition of } \omega^\alpha \cdot x \\ & = & \underbrace{g_{\omega^\alpha}(\cdots(g_{\omega^\alpha}}_{x \text{ times}}(x))\cdots) & \text{by Property 1 above} \\ & = & g_{\omega^\alpha}^{(x)}(x). & \end{array}$$

Part 1 now follows by induction on n. For $n = 0$,

$$\begin{array}{rcll} g_{\omega^0}(x) & = & g_1(x) & \text{by definition of } \omega^0 \\ & = & g_0(x+1) & \text{by definition of } g_1 \\ & = & x+1 & \text{by definition of } g_0 \\ & = & h_0(x) & \text{by definition of } h_0. \end{array}$$

For $n+1$,

$$\begin{array}{rcll} g_{\omega^{n+1}}(x) & = & g_{\omega^n}^{(x)}(x) & \text{by Property 2 above} \\ & = & h_n^{(x)}(x) & \text{by induction hypothesis} \\ & = & h_{n+1}(x) & \text{by definition of } h_{n+1}. \end{array}$$

Part 2 follows from Part 1, as follows:

$$\begin{array}{rcll} g_{\omega^\omega}(x) & = & g_{(\omega^\omega)_x}(x) & \text{by definition of } g_{\omega^\omega} \\ & = & g_{\omega^x}(x) & \text{by definition of } (\omega^\omega)_x \\ & = & h_x(x) & \text{by Part 1} \\ & = & d_{\mathcal{P}_1}(x) & \text{by definition of } d_{\mathcal{P}_1}. \quad \Box \end{array}$$

Exercises VIII.8.24 A refinement of the Grzegorczyk Hierarchy. (Wainer [1972]) The first idea is to use the functions g_α's in place of the h_n's in VIII.8.12. Since many of these functions are elementarily equivalent, one cannot simply use $\mathcal{E}(g_\alpha)$ as a class of the new hierarchy. By VIII.7.4, $\mathcal{E}$ is obtained by closure under composition and bounded primitive recursion. The second idea is then to use bounded composition as well, where we say that f is defined from $t_1, \dots, t_m$, h and t by **bounded composition** if

$$f(\vec{x}) = h(t_1(\vec{x}), \dots, t_m(\vec{x})) \qquad \text{and} \qquad f(\vec{x}) \leq t(\vec{x}).$$

We now start from $\mathcal{G}_0 = \emptyset$, and let $\mathcal{G}_{\alpha+1}$ be the smallest class of functions containing $\mathcal{G}_\alpha$, the initial functions and g_α, and closed under bounded composition and bounded primitive recursion. Finally, $\mathcal{G}_\alpha = \bigcup_{\beta<\alpha} \mathcal{G}_\beta$ when α is a limit ordinal.

a) *$\mathcal{G}_\alpha \subset \mathcal{G}_\beta$, if $2 \leq \alpha < \beta < \omega^\omega$.* (Hint: every function in $\mathcal{G}_{\alpha+1}$ is bounded by g_α by definition, and so it is enough to show that g_β strictly dominates g_α. This can be proved by induction on β, showing simultaneously that g_β is strictly increasing. For β a limit, it is enough to show that $g_{\beta_n}(x) \leq g_{\beta_{n+1}}(x)$, for every n and x.)

b) *$\mathcal{G}_{\omega^n} = \mathcal{E}_n$, if $n \geq 3$.* (Hint: using VIII.8.23, show that $\mathcal{G}_{\omega^{n+1}} = \mathcal{E}(h_n)$, for $n \geq 2$. $\mathcal{G}_{\omega^{n+1}}$ is closed under bounded primitive recursion because so is every $\mathcal{G}_{\alpha+1}$, by definition. To show closure under full composition, let f be defined by composition from $t_1, \dots, t_m$ and h in $\mathcal{G}_{\omega^{n+1}}$. By induction hypothesis, there is a z such that $\alpha = \omega^n \cdot z$, and t_i and h are all in $\mathcal{G}_\alpha$. Then $f(\vec{x}) \leq g_\alpha(\sum_{1\leq i\leq m} t_i(\vec{x}))$. For a large enough α, the class $\mathcal{G}_\alpha$ is closed under sum. Thus $\sum_{1\leq i\leq m} t_i(\vec{x}) \leq g_\alpha(\sum \vec{x})$ and

$$f(\vec{x}) \leq g_\alpha(g_\alpha(\sum \vec{x})) = g_{\alpha\cdot 2}(\sum \vec{x}).$$

This shows that $\mathcal{E}(h_n) \subseteq \mathcal{G}_{\omega^{n+1}}$. The converse is proved by showing, by induction on $\alpha < \omega^{n+1}$, that g_α is elementary in h_n.)

c) *$\mathcal{G}_{\alpha+1}$ can be defined as the smallest class of functions containing the initial functions and g_α, and closed under bounded composition and bounded primitive recursion.* (Hint: to show that $\mathcal{G}_\alpha \subseteq \mathcal{G}_{\alpha+1}$, it is enough to show by induction on $\beta < \alpha$ that $g_\beta \in \mathcal{G}_\alpha$. If $\beta < \omega^3$, then $\beta = \omega^2 \cdot m + \omega \cdot n + k$ and, by the properties proved in the proof of VIII.8.23, $g_\beta(z) = g_{\omega^2}^{(m)}(g_\omega^{(n)}(g_1^{(k)}(z)))$. Then g_β is defined by composition and primitive recursion bounded by g_α. If $\omega^3 \leq \beta$ and $\alpha = \beta + n$, we obtain the result by descending induction. For $\omega^3 < \beta + \omega \leq \alpha$, recall from VIII.7.6 that any function f is the composition of a fixed elementary function (which is in $\mathcal{G}_\beta$, since $\omega^3 \leq \beta$) and any computation time for f. It is now enough to show that $g_\beta(x)$ is computable in time $g_\beta(x + c_\beta)$ for some constant c_β, since then $g_\beta(x + c_\beta) = g_{\beta + c_\beta}(x) \leq g_\alpha(x)$, and g_β is definable by bounded composition in $\mathcal{G}_\alpha$.)

We now turn to the finer Slow Growing Hierarchy, with the goal of computing the position of $d_{\mathcal{P}_1}$ in it. We might expect to go a long way, since it takes ϵ_0 steps to get to $d_{\mathcal{E}}$, which is a function elementarily equivalent to h_3.

The intuition that takes us to the ordinal we want is the following. To obtain $f_{\epsilon_0} \equiv_{\mathcal{E}} h_3$ we started from the exponential function $f_{\omega^\omega} \equiv_{\mathcal{E}} h_2$, and iterated it. The ordinal ϵ_0 measured the number of steps needed to complete the iteration, i.e. how long it takes to obtain a fixed-point for the ordinal exponential function ω^α. Since h_{n+1} is the iteration of h_n, to obtain h_4 we have to iterate the process that took us to ϵ_0, and obtain its first fixed-point. We are thus led to the following definition.

Definition VIII.8.25 The Veblen Hierarchy (Veblen [1908]) *The sequence $\{\phi_n\}_{n\in\omega}$ of ordinal functions is defined by induction on n, as follows:*

$$\begin{array}{rcl} \phi_0(\gamma) & = & \omega^\gamma \\ \phi_{n+1}(\gamma) & = & \textit{the } \gamma\textit{-th fixed-point of } \phi_n, \\ & = & \textit{the } \gamma\textit{-th ordinal } \beta \textit{ such that } \beta = \phi_n(\beta). \end{array}$$

Moreover,

$$\phi_\omega(0) = \lim_{x\in\omega} \phi_x(0).$$

We can spell out the definition just given, as follows:

- $\phi_{n+1}(0) = \lim_{x\in\omega} \phi_n^{(x)}(0)$
  This can be seen by noticing that ϕ_n is monotone and continuous. Indeed, $0 \leq \phi_{n+1}(0)$, and by monotonicity

  $$\phi_n(0) \leq \phi_n(\phi_{n+1}(0)) = \phi_{n+1}(0),$$

  since $\phi_{n+1}(0)$ is a fixed-point of ϕ_n.

  In general, $\phi_n^{(x)}(0) \leq \phi_{n+1}(0)$, and so $\lim_{x\in\omega} \phi_n^{(x)}(0) \leq \phi_{n+1}(0)$. Moreover, $\lim_{x\in\omega} \phi_n^{(x)}(0)$ is a fixed-point of ϕ_n by continuity:

  $$\phi_n(\lim_{x\in\omega} \phi_n^{(x)}(0)) = \lim_{x\in\omega} \phi_n(\phi_n^{(x)}(0)) = \lim_{x\in\omega} \phi_n^{(x+1)}(0) = \lim_{x\in\omega} \phi_n^{(x)}(0).$$

- $\phi_{n+1}(\gamma+1) = \lim_{x\in\omega} \phi_n^{(x)}(\phi_{n+1}(\gamma)+1)$
  Similarly, since $\phi_{n+1}(\gamma+1)$ is the smallest fixed-point of ϕ_n greater than $\phi_{n+1}(\gamma)$.

- $\phi_{n+1}(\gamma) = \lim_{x\in\omega} \phi_{n+1}(\gamma_x)$, *if γ is a limit*
  By continuity of ϕ_n.

Notice that every ordinal $\alpha \neq 0$ has a unique expansion of the form

$$\alpha = \phi_{n_1}(\beta_1)\cdot x_1 + \cdots + \phi_{n_m}(\beta_m)\cdot x_m,$$

where $x_1,\dots,x_m$ are integers, $\phi_{n_m}(\beta_m) < \cdots < \phi_{n_1}(\beta_1)$ and $n_m \leq \cdots \leq n_1$.

Notice also that $\phi_\omega(0)$ is the smallest α such that $\alpha = \phi_n(\alpha)$ for every n. If $\alpha < \phi_\omega(0)$, then in the above expansion of α we have $\phi_{n_1}(\beta_1) < \alpha$, and we can use the expansion in inductive definitions. From this representation, it follows that every limit ordinal $\alpha < \phi_\omega(0)$ can be uniquely written in one of the forms

$$\beta + \phi_n(0),\ n \neq 0 \qquad \text{or} \qquad \beta + \phi_n(\gamma+1) \qquad \text{or} \qquad \beta + \phi_n(\gamma),\ \gamma \text{ a limit}$$

with $\gamma < \alpha$. This justifies the next definition, which extends VIII.7.13 (since $\phi_1(0) = \epsilon_0$).

Definition VIII.8.26 Fundamental Sequences up to $\boldsymbol{\phi_\omega(0)}$. *If $\alpha < \phi_\omega(0)$ is a limit ordinal, then*

$$\alpha_x = \begin{cases} \omega^\gamma \cdot x & \textit{if } \alpha = \omega^{\gamma+1} = \phi_0(\gamma+1) \\ \omega^{\gamma_x} & \textit{if } \alpha = \omega^\gamma = \phi_0(\gamma),\ \gamma \textit{ a limit} \\ \phi_n^{(x)}(0) & \textit{if } \alpha = \phi_{n+1}(0) \\ \phi_n^{(x)}(\phi_{n+1}(\gamma)+1) & \textit{if } \alpha = \phi_{n+1}(\gamma+1) \\ \phi_{n+1}(\gamma_x) & \textit{if } \alpha = \phi_{n+1}(\gamma),\ \gamma \textit{ a limit} \\ \beta + \gamma_x & \textit{if } \alpha = \beta+\gamma,\ \gamma \textit{ a limit.} \end{cases}$$

Moreover,

$$(\phi_\omega(0))_x = \phi_x(0).$$

In the following, we use the fundamental sequences just defined in the definition VIII.7.12 of f_α, when $\alpha \leq \phi_\omega(0)$.

Since $\phi_1(0) = \epsilon_0$ and $h_3 \equiv_{\mathcal{E}} d_{\mathcal{E}}$, VIII.7.14 proves that $h_3 \equiv_{\mathcal{E}} f_{\phi_1(0)}$. This is generalized in Part 1 of the next result.

Proposition VIII.8.27 The Position of $\boldsymbol{d_{\mathcal{P}_1}}$ in the Slow Growing Hierarchy (Girard [1981])

1. *$h_{n+2} \equiv_{\mathcal{E}} f_{\phi_n(0)}$, for every n*

2. $d_{\mathcal{P}_1} \equiv_{\mathcal{E}} f_{\phi_\omega(0)}$.

Proof. The basic observation is that the definition of the finite levels of the Veblen Hierarchy is similar to the definition of the functions h_n, according to the following correspondence:

ordinals	numbers
ordinal exponentiation	numerical exponentiation
fixed-points	iteration

It is thus enough to define functions H_n mimicking the definition of the Veblen Hierarchy, and prove that they are elementarily equivalent to the functions h_n.

We thus define

$$\begin{array}{rcl} H_0(x,y) & = & x^y \\ H_{n+1}(x,0) & = & H_n^{(x)}(x,0) \\ H_{n+1}(x,y+1) & = & H_n^{(x)}(x,H_{n+1}(x,y)+1). \end{array}$$

Here the iteration of H_n is performed on its second variable:

$$\begin{array}{rcl} H_n^{(0)}(x,y) & = & y \\ H_n^{(z+1)}(x,y) & = & H_n(x,H_n^{(z)}(x,y)). \end{array}$$

It is quite clear that $H_n(x,0)$ and $h_{n+2}(x)$ are elementarily equivalent, since $H_0(x,0)=1$ is elementary (as is h_2), and H_{n+1} is obtained from H_n by iteration (notice that H_0 is of exponential growth, as is h_2). We now show that, for $\phi_n(\beta) < \phi_\omega(0)$,

$$f_{\phi_n(\beta)}(x) \quad = \quad H_n(x, f_\beta(x)), \qquad \text{(VIII.5)}$$

so that

$$f_{\phi_n(0)}(x) = H_n(x,0).$$

This proves Part 1, from which Part 2 follows by uniformity.

The proof of VIII.5 is quite immediate, since the definition of H_n mimics the definition of the fundamental sequences. We proceed by induction on n. For $n=0$,

$$\begin{array}{rcll} f_{\phi_0(\beta)}(x) & = & f_{\omega^\beta}(x) & \text{by definition of } \phi_0 \\ & = & x^{f_\beta(x)} & \text{as in Part 2 of the proof of VIII.7.14} \\ & = & H_0(x,f_\beta(x)) & \text{by definition of } H_0. \end{array}$$

For $n+1$, we proceed by induction on β:

- $\beta = 0$

$$\begin{array}{rcll} f_{\phi_{n+1}(0)}(x) & = & f_{(\phi_{n+1}(0))_x}(x) & \text{by definition of } f_{\phi_{n+1}(0)} \\ & = & f_{\phi_n^{(x)}(0)}(x) & \text{by definition of } (\phi_{n+1}(0))_x \\ & = & H_n^{(x)}(x, f_0(x)) & \text{by induction hypothesis on } n \\ & = & H_n^{(x)}(x, 0) & \text{by definition of } f_0 \\ & = & H_{n+1}(x, 0) & \text{by definition of } H_{n+1} \\ & = & H_{n+1}(x, f_0(x)) & \text{by definition of } f_0. \end{array}$$

- $\beta = \gamma + 1$

$$\begin{array}{rll} & f_{\phi_{n+1}(\gamma+1)}(x) & \\ = & f_{(\phi_{n+1}(\gamma+1))_x}(x) & \text{by definition of } f_{\phi_{n+1}(\gamma+1)} \\ = & f_{\phi_n^{(x)}(\phi_{n+1}(\gamma)+1)}(x) & \text{by definition of } (\phi_{n+1}(\gamma+1))_x \\ = & H_n^{(x)}(x, f_{\phi_{n+1}(\gamma)+1}(x)) & \text{by induction hypothesis on } n \\ = & H_n^{(x)}(x, f_{\phi_{n+1}(\gamma)}(x) + 1) & \text{by definition of } f_{\phi_{n+1}(\gamma)+1} \\ = & H_n^{(x)}(x, H_{n+1}(x, f_\gamma(x)) + 1) & \text{by induction hypothesis on } \beta \\ = & H_{n+1}(x, f_\gamma(x) + 1) & \text{by definition of } H_{n+1} \\ = & H_{n+1}(x, f_{\gamma+1}(x)) & \text{by definition of } f_{\gamma+1}. \end{array}$$

- β *a limit*

$$\begin{array}{rcll} f_{\phi_{n+1}(\beta)}(x) & = & f_{(\phi_{n+1}(\beta))_x}(x) & \text{by definition of } f_{\phi_{n+1}(\beta)} \\ & = & f_{\phi_{n+1}(\beta_x)}(x) & \text{by definition of } (\phi_{n+1}(\beta))_x \\ & = & H_{n+1}(x, f_{\beta_x}(x)) & \text{by induction hypothesis on } \beta \\ & = & H_{n+1}(x, f_\beta(x)) & \text{by definition of } f_\beta. \end{array}$$

It only remains to prove Part 2. Clearly

$$H_\omega(x, x) = H_x(x, 0)$$

is elementarily equivalent to $d_{\mathcal{P}_1}$. And

$$\begin{array}{rcll} f_{\phi_\omega(0)}(x) & = & f_{(\phi_\omega(0))_x}(x) & \text{by definition of } f_{\phi_\omega(0)} \\ & = & f_{\phi_x(0)}(x) & \text{by definition of } (\phi_\omega(0))_x \\ & = & H_x(x, f_0(x)) & \text{by VIII.5} \\ & = & H_x(x, 0) & \text{by definition of } f_0 \\ & = & H_\omega(x, x) & \text{by definition of } H_\omega. \quad \Box \end{array}$$

VIII.9 ϵ_0-Recursive Functions

In this section we go beyond the class of primitive recursive functions studied in the last section, and extend the notion of primitive recursion from ω to limit ordinals α. If α is closed under ordinal exponentiation (for example, if $\alpha = \epsilon_0$), the class of functions definable by primitive recursion on α resembles the class of primitive recursion functions in many ways, and this provides a number of interesting subrecursive classes.

The work in this section relies pretty heavily on ordinals and operations on them, and we refer to Monk [1969] and Levy [1979] for fundamentals on the subject. However, the section is mostly self-contained, and requires only the working knowledge of the elementary Set Theory used until now.

Multiple recursive functions ⋆

A natural method of extending the class of primitive recursive functions comes from VIII.8.11, which shows that nested recursion on *two* variables produces the diagonal function $d_{\mathcal{P}_1}$.

Definition VIII.9.1 (Peter [1935]) *For $n \geq 1$, the class* $\boldsymbol{\mathcal{P}_n}$ *of* ***n*-recursive functions** *is the smallest class of functions:*

1. *containing the initial functions*
2. *closed under composition*
3. *closed under nested recursion on at most n variables.*

The class $\boldsymbol{\mathcal{P}_\omega}$ *of* **multiple recursive functions** *is* $\bigcup_{n\in\omega} \mathcal{P}_n$.

VIII.8.5 and VIII.8.11 show that for $n = 1$, we obtain the class of primitive recursive functions $\mathcal{P}_1$ (whence the name $\mathcal{P}_n$), and that $\mathcal{P}_1 \subset \mathcal{P}_2$.

Exercise VIII.9.2 *$\mathcal{P}_n \subset \mathcal{P}_{n+1}$, for each $n \geq 1$.* (Peter [1936]) (Hint: $\mathcal{P}_{n+1}$ contains a universal function for $\mathcal{P}_n$, which can be defined by nested recursion on one more variable for the indices.)

We do not pursue here the study of recursion on many variables, since we prefer the more general approach of ordinal recursion, which subsumes the present one as a special case (see VIII.9.7).

For more results on n-recursive functions see Peter [1951], Axt [1959], [1961], Lachlan [1962], Robbin [1965], Löb and Wainer [1970a], Marchenkov [1970], [1972], and Schwichtenberg [1972].

Programs for the n-recursive functions are studied in Constable [1971a], Machtey [1972], and S. Muchnik [1976a].

Ordinal recursion

To illustrate the main idea used in this section, we consider the function defined by

$$\begin{aligned} h(0,z) &= z+1 \\ h(n+1,0) &= h(n,1) \\ h(n+1,z+1) &= h(n,h(n+1,z)), \end{aligned}$$

which is a simplified version (with similar properties) of the function h described in VIII.8.11.

We concentrate on the arguments n and z, which we can arrange in a doubly infinite matrix consisting of the pairs (n,z). The values for the arguments in the first row ($n=0$) are given by a known function (of z). The values for arguments in other rows ($n>0$) are given in terms of values of the function itself for arguments either preceding the given one in the same row, or in rows preceding the given one. For example, for the argument $(n+1,z+1)$ we need both the value of h for the previous argument $(n+1,z)$ in the same row, and the value for the argument $(n,h(n+1,z))$ in the previous row. In other words, the definition of h is by induction on the lexicographical ordering of pairs, which corresponds to a well-ordering of ordinal ω^2.

We are thus led to the following extensions of the notions of unnested and nested primitive recursion (VIII.8.1 and VIII.8.4).

Definition VIII.9.3 Recursion on Well-Orderings and Well-Founded Relations (Hilbert [1926], Ackermann [1940]) *Let $\prec$ be a well-ordering of the natural numbers, with 0 as the least element.*

The class of **$\prec$-recursive functions** *is the smallest class of functions:*

1. *containing the initial functions*
2. *closed under composition*
3. *closed under primitive recursion*
4. *closed under the following schema of* **$\prec$-recursion***:*

$$\begin{aligned} f(\vec{x},0) &= g(\vec{x}) \\ f(\vec{x},y+1) &= h(\vec{x},y,f(\vec{x},q(\vec{x},y+1))), \end{aligned}$$

where q is a function such that:

$$q(\vec{x},0)=0 \qquad \textit{and} \qquad q(\vec{x},y+1) \prec y+1$$

(q is more than a predecessor function for $\prec$, since $y+1$ might correspond to a limit ordinal).

The class of **nested $\prec$-recursive functions** *is the smallest class of functions:*

1. *containing the initial functions*
2. *closed under composition*
3. *closed under primitive recursion*
4. *closed under the following schema of* **nested $\prec$-recursion***:*

$$\begin{array}{rcl} f(\vec{x},0) & = & g(\vec{x}) \\ f(\vec{x},y+1) & = & h(\vec{x},y), \end{array}$$

where $h(\vec{x},y)$ is a numerical term built up from:

- *natural numbers n*
- *the variables $\vec{x}$ and y*
- *a finite number of known functions $g_1,\ldots,g_m$*
- *the functional letter f*
- *the function p (for 'predecessor w.r.t. $\prec$'), defined as:*

$$p(a,b)=\begin{cases} a & \textit{if } a\prec b \\ 0 & \textit{otherwise.}\end{cases}$$

Moreover, $h(\vec{x},y)$ must be built in such a way that f appears in it only in contexts of the form $f(\vec{t},p(s,y+1))$, where $\vec{t}$ and s are terms.

The next exercise shows that, for the goal of classifying the recursive functions, neither the ordinal nor the complexity of $\prec$ tells much about the complexity of a $\prec$-recursive function.

Exercise VIII.9.4 *Every recursive function is $\prec$-recursive w.r.t. some polynomial time computable well-ordering $\prec$ of ordinal ω, with a polynomial time computable predecessor function.* (Myhill [1953a], Routledge [1953], Liu [1960a], Fabian and Kent [1969]) (Hint: consider a total recursive function $f(x)=\mathcal{U}(\mu y\,\mathcal{T}_1(e,x,y))$, so that $(\forall x)(\exists y)\mathcal{T}_1(e,x,y)$. If $\prec$ is an ordering of code numbers of pairs such that

$$\langle \mu y\,\mathcal{T}_1(e,x,y),x\rangle \prec \langle \mu y\,\mathcal{T}_1(e,x,y)-1,x\rangle \prec\cdots\prec\langle 1,x\rangle\prec\langle 0,x\rangle$$

and D is its domain, we define a $\prec$-recursive function g such that $f(x)=g(\langle 0,x\rangle)$ as follows:

$$g(\langle z,x\rangle)=\begin{cases} \mathcal{U}(z) & \text{if } \langle z,x\rangle\in D \text{ and } \langle z+1,x\rangle\notin D \\ g(\langle z+1,x\rangle) & \text{if } \langle z+1,x\rangle\in D \\ 0 & \text{if } \langle z,x\rangle\notin D.\end{cases}$$

The domain of D consists of all pairs $\langle z, x\rangle$ such that $z \leq \mu y\, \mathcal{T}_1(e,x,y)$, i.e. such that $(\forall t < z)\neg\mathcal{T}_1(e,x,t)$. If $\langle a, x\rangle$ and $\langle b, y\rangle$ are both in D, then

$$\langle a,x\rangle \prec \langle b,y\rangle \;\Leftrightarrow\; x < y \;\vee\; (x = y \wedge b < a)$$

defines $\prec$. By the Improved Normal Form Theorem VIII.2.8, $\prec$ is polynomial time computable.)

ϵ_0-recursive functions

The usual solution to override the obstacle described in the exercise above is to restrict the class of well-orderings to 'natural' ones.

Definition VIII.9.5 Canonical Well-Orderings up to ϵ_0 (Hilbert and Bernays [1939]) *Let*

$$\omega(0) = 1 \qquad \textit{and} \qquad \omega(n+1) = \omega^{\omega(n)}.$$

By induction on $n \geq 1$, we define

$\prec_n$	*a well-ordering of ω of ordinal $\omega(n)$*
$ord_n(x)$	*the ordinal associated with x in $\prec_n$*
$num_n(\alpha)$	*the number associated with $\alpha < \omega(n)$ in $\prec_n$*

as follows:

1. *for $n = 1$*
   - $\prec_1 = <$
   - $ord_1(x) = x$
   - $num_1(\alpha) = \alpha$.
2. *for $n + 1$*

   Suppose $\prec_n$, ord_n and num_n are given. Then:

   - *If $\alpha = \omega^{\beta_1}\cdot x_1 + \cdots + \omega^{\beta_m}\cdot x_m$, where $x_1, \ldots, x_m$ are positive integers and $\beta_m < \cdots < \beta_1$ (the Cantor normal form of α), then*

   $$num_{n+1}(\alpha) = p^{x_m}_{num_n(\beta_m)} \cdots p^{x_1}_{num_n(\beta_1)} - 1.$$

   - *If $b_m \prec_n \cdots \prec_n b_1$ and $x = p^{x_m}_{b_m} \cdots p^{x_1}_{b_1} - 1$, then*

   $$ord_{n+1}(x) = \omega^{ord_n(b_1)} \cdot x_1 + \cdots + \omega^{ord_n(b_m)} \cdot x_m.$$

- *The ordering $\prec_{n+1}$ on numbers is induced by the ordering on the ordinals associated with them:*

$$x \prec_{n+1} y \Leftrightarrow ord_{n+1}(x) < ord_{n+1}(y).$$

Notice that every ordinal $\alpha < \omega(n)$ has a unique notation $num_n(\alpha)$ in $\prec_n$, every number x denotes an ordinal $ord_n(x) < \omega(n)$, and

$$num_n(ord_n(x)) = x \qquad \text{and} \qquad ord_n(num_n(\alpha)) = \alpha.$$

Also, 0 is the least element of $\prec_n$, for every n.

Definition VIII.9.6 (Kreisel [1952]) *For any limit ordinal $\alpha < \epsilon_0$, the classes* $\boldsymbol{U(\alpha)}$ *and* $\boldsymbol{N(\alpha)}$ *of* **unnested** *and* **nested $\boldsymbol{\alpha}$-recursive functions** *are the classes of unnested and nested $\prec_\alpha$-recursive functions, where $\prec_\alpha$ is the lower cut of $\prec_n$ defined by $num_n(\alpha)$, for the smallest n such that $\alpha < \omega(n)$.*

The classes $\boldsymbol{U(\epsilon_0)}$ *and* $\boldsymbol{N(\epsilon_0)}$ *of* **unnested** *and* **nested $\boldsymbol{\epsilon_0}$-recursive functions** *are defined as:*

$$U(\epsilon_0) = \bigcup_{\alpha<\epsilon_0} U(\alpha) \qquad \textit{and} \qquad N(\epsilon_0) = \bigcup_{\alpha<\epsilon_0} N(\alpha),$$

i.e. a function is (nested) ϵ_0-recursive if it (nested) α-recursive, for some limit ordinal $\alpha < \epsilon_0$.

The following properties are immediate:

- $U(\omega) = N(\omega)$ is the class of primitive recursive functions (by VIII.8.5)
- $U(\alpha) \subseteq N(\alpha)$, for $\alpha \leq \epsilon_0$
- $U(\alpha) \subseteq U(\beta)$ and $N(\alpha) \subseteq N(\beta)$, for $\alpha \leq \beta \leq \epsilon_0$ (because there is a primitive recursive similarity map of $\prec_\alpha$ onto an initial segment of $\prec_\beta$).

Exercise VIII.9.7 *For $n \geq 1$, $\mathcal{P}_n = N(\omega^n)$.* (Ackermann [1940], Peter [1950]) (Hint: identify in a primitive recursive way $\prec_{\omega^n}$ and the lexicographical order of n-tuples of integers.)

Closure properties

The next result is the appropriate generalization of VIII.8.5.

Theorem VIII.9.8 Nested Ordinal Recursion (Tait [1961]) *For any limit ordinal α such that $\omega \leq \alpha < \epsilon_0$, $N(\alpha) \subseteq U(\omega^\alpha)$.*

Proof. We only indicate the modifications to be made in the proof of VIII.8.5. Using the same notation, we let

$$p_\beta(a,b) = \begin{cases} a & \text{if } a \prec_\beta b \\ 0 & \text{otherwise} \end{cases}$$

for any limit ordinal $\beta < \epsilon_0$ (where $\prec_\beta$ is defined as in VIII.9.6).

Let f be defined by nested α-recursion from g and h, i.e.

$$\begin{aligned} f(\vec{x},0) &= g(\vec{x}) \\ f(\vec{x},y+1) &= h(\vec{x},y), \end{aligned}$$

where $h(\vec{x},y)$ contains only f-terms of the form $f(\vec{t}, p_\alpha(s,y+1))$.

The primitive recursive functions and predicate ev, red and R are defined as in VIII.8.5. The main difference with the proof of VIII.8.5 is that the function β such that, when x codes a constant normal term,

$$\beta(x) = \mu z\,(red^{(z)}(x) \text{ is proper})$$

must now be defined by unnested ω^α-recursion (as opposed to primitive recursion as in VIII.8.5).

We assign ordinal notations $ord(x)$ for ordinals less than ω^α (as opposed to finite ordinals as in VIII.8.5) to codes of normal terms x, in such a way that ord is ω^α-recursive, and

$$R(x) \;\Rightarrow\; ord(red(x)) \prec_{\omega^\alpha} ord(x),$$

i.e. the ordinals are assigned in such a way that they decrease when a reducible term is reduced. Using this we will be able to perform an ω^α-recursion on the notations. In other words, we can define $\gamma(x,y)$ (where y should be thought of as $ord(x)$), as

$$\begin{aligned} \gamma(x,0) &= 0 \\ \gamma(x,y+1) &= \begin{cases} 0 & \text{if } \neg R(x) \\ 1+\gamma(red(x), p_{\omega^\alpha}(ord(red(x)),y+1)) & \text{otherwise.} \end{cases} \end{aligned}$$

Unlike the case of VIII.8.5, γ is automatically ω^α-recursive (even if red is substituted in place of the parameter x), because if $z = p_{\omega^\alpha}(ord(red(x)),y+1)$, then $z \prec_{\omega^\alpha} y+1$, and so the pair $(red(x),z)$ precedes the pair $(x,y+1)$ in the standard antilexicographical[14] well-ordering of pairs, of ordinal $\omega \cdot \omega^\alpha$, and

[14]In the *lexicographical* order of pairs the leading component is the first one, i.e.

$$(a,b) < (a',b') \;\Leftrightarrow\; (a < a') \,\vee\, (a = a' \,\wedge\, b < b').$$

In the *antilexicographical* order the leading component is instead the second one, i.e.

$$(a,b) < (a',b') \;\Leftrightarrow\; (b < b') \,\vee\, (a < a' \,\wedge\, b = b').$$

$\omega \cdot \omega^\alpha = \omega^\alpha$ for $\alpha \geq \omega$. In other words, γ is automatically ω^α-recursive, if $\alpha \geq \omega$.

It only remains to assign the ordinals $ord(x)$. Let r be the number of all f-terms occurring in $h(\vec{x}, y)$.

- if $x = \langle u \rangle$ and u is not normal, let
$$ord(x) = 0$$
- if $x = \langle u \rangle$ and u is normal, consider the occurrences $f_1, \ldots, f_s$ of f-terms in it and, if $\beta_1, \ldots, \beta_s$ are the ordinals associated with them, let
$$ord(x) = \beta_1 + \cdots + \beta_s$$
- if $x = \langle f(\vec{t}, p_\alpha(s, n+1)) \rangle$, i.e. x codes a normal f-term, we let
$$ord(x) = \begin{cases} \omega^{ord_\alpha(n+1)} & \text{if } s \text{ is not proper} \\ \omega^{ev(\langle\, p_\alpha(s,n+1)\rangle)} \cdot r & \text{otherwise,} \end{cases}$$
where $ord_\alpha(n+1)$ is the ordinal denoted by $n+1$ in $\prec_\alpha$. The reason is the following. In the first case we cannot evaluate the term directly, so we just indicate that our term may be used in the computation of $f(\vec{x}, n+1)$. In the second case we can instead eliminate our term, according to the reduction rules, in favor of r new occurrences of f-terms, each of them used to compute a value of the form $f(\vec{t_j}, ev(\langle\, p_\alpha(s_j, n+1)\rangle))$.

Notice how the ordinal assignment is slightly simpler here than in VIII.8.5 (in particular, it does not need the additional function δ introduced there, whose role is now taken by ordinal exponentiation), since we are not trying to keep the ordinals less than α itself, and we content ourselves with ordinals less than ω^α.

The proof that
$$R(x) \;\Rightarrow\; ord(red(x)) \prec_{\omega^\alpha} ord(x)$$
is similar to (but simpler than) that in VIII.8.5, using standard properties of ordinal exponentiation. □

Corollary VIII.9.9 (Kreisel [1952]) $U(\epsilon_0) = N(\epsilon_0)$.

Proof. An unnested ϵ_0-recursive function is obviously nested ϵ_0-recursive. Conversely, a nested ϵ_0-recursive function is nested α-recursive for some $\alpha < \epsilon_0$ by definition, hence unnested ω^α-recursive by VIII.9.8, and so unnested ϵ_0-recursive because $\omega^\alpha < \epsilon_0$ (by definition of ϵ_0, since $\alpha < \epsilon_0$). □

Exercises VIII.9.10 a) *$N(\omega \cdot \alpha) \subseteq U(\omega^\alpha)$, for any $\alpha \leq \epsilon_0$.* (Wainer [1972]) (Hint: in the proof of VIII.9.8, if r is the number of all f-terms occurring in $h(\vec{x}, y)$, replace ω by $r+1$ as base of the ordinal assignment, so that $r+1 \geq 2$. Then nested $\omega \cdot \alpha$-recursion is reduced to unnested $(r+1)^{\omega \cdot \alpha} = \omega^\alpha$-recursion.)

b) *$U(\omega^\alpha) \subseteq N(\omega \cdot \alpha)$, for any $\alpha \leq \epsilon_0$.* (Tait [1961]) (Hint: the proof of VIII.9.12 will show that it is enough, given any ω^α-recursive function q such that $q(\vec{x}, 0) = 0$ and $q(\vec{x}, y+1) \prec_{\omega^\alpha} y+1$, to obtain an $\omega \cdot \alpha$-recursive function $t(\vec{x}, y)$ giving the number of steps required to get to 0 when starting from $(\vec{x}, y)$, by repeated applications of q. The idea is to look at notations in the standard ordering of ω^α, i.e. numbers of the form $p_{b_m}^{x_m} \cdots p_{b_1}^{x_1}$, and to consider the antilexicographical well-ordering of pairs $\langle a_i, b_i \rangle$ of ordinal $\omega \cdot \alpha$.)

From the previous results and exercises we have that, for $\alpha \leq \epsilon_0$,

$$N(\alpha) \subseteq N(\omega \cdot \alpha) = U(\omega^\alpha).$$

The inclusion here (and hence in VIII.9.8) may be proper. For example, $N(\omega)$ and $N(\omega^2)$ are respectively $\mathcal{P}_1$ and $\mathcal{P}_2$, and hence $N(\omega) \subset N(\omega^2)$ by VIII.8.11. More generally, $N(\omega^n) \subset N(\omega^{n+1})$ for every n, by VIII.9.2 and VIII.9.7.

Equality may also hold, as VIII.9.9 shows. For other examples, let $\alpha = \omega(n)$ and $n > 1$. Then $\alpha = \omega \cdot \alpha$, and thus $N(\omega(n)) = U(\omega(n+1))$, for every $n > 1$.

The restriction to $\alpha \leq \epsilon_0$ is not necessary in the results above. For example, Tait [1961] shows that they hold for any countable ordinal α, almost independently (i.e. modulo some regularity conditions) of the well-ordering $\prec_\alpha$ chosen to represent it, as long as the well-ordering $\prec_{\omega^\alpha}$ chosen to represent ω^α is obtained from $\prec_\alpha$ in the same way as $\prec_{n+1}$ was obtained from $\prec_n$ in VIII.9.5.

For more on abstract ordinal recursion, see Peter [1951], Tait [1961], Kent [1969], Löb and Wainer [1970], [1970a], [1971], Schwichtenberg [1972], Fairtlough and Wainer [1992], Möllerfeld and Weiermann [199?].

Alternative characterizations

In Section 8 we have been able to obtain a hierarchy for the primitive recursive functions by relying on VIII.8.7, which reduced the general schema of primitive recursion to the more manageable one of iteration. The next definition and result provide an analogue for the ordinal recursive functions.

Definition VIII.9.11 *Let $\prec_\alpha$ be defined as in VIII.9.6, and let q be a function such that*

$$q(\vec{x}, 0) = 0 \qquad \textit{and} \qquad q(\vec{x}, y+1) \prec_\alpha y+1.$$

The **α-annihilation** *of q is the function t defined by*

$$\begin{aligned} t(\vec{x},0) &= 0 \\ t(\vec{x},y+1) &= 1+t(\vec{x},q(\vec{x},y+1)). \end{aligned}$$

Thus t counts the number of steps needed to get to 0 when starting from $(\vec{x},y)$, by repeated applications of the descending function q.

Proposition VIII.9.12 (Tait [1961], Robbin [1965]) *For any limit ordinal $\alpha < \epsilon_0$, $U(\alpha)$ is the smallest class of functions:*

1. *containing the initial functions*
2. *closed under composition*
3. *closed under primitive recursion*
4. *closed under α-annihilation.*

Proof. Let $\mathcal{C}$ be the smallest class satisfying the stated conditions: clearly every function in $\mathcal{C}$ is α-recursive, since α-annihilation is a special case of $\prec_\alpha$-recursion.

For the converse, we only have to show that $\mathcal{C}$ is closed under $\prec_\alpha$-recursion. Let g, h and q be functions in $\mathcal{C}$, and

$$\begin{aligned} f(\vec{x},0) &= g(\vec{x}) \\ f(\vec{x},y+1) &= h(\vec{x},y,f(\vec{x},q(\vec{x},y+1))), \end{aligned}$$

where

$$q(\vec{x},0)=0 \qquad \text{and} \qquad q(\vec{x},y+1) \prec_\alpha y+1.$$

We first define, by primitive recursion on z, a function $f_1(\vec{x},y,z)$ giving the result of z steps in the calculation of f:

$$\begin{aligned} f_1(\vec{x},y,0) &= g(\vec{x}) \\ f_1(\vec{x},y,z+1) &= \begin{cases} g(\vec{x}) & \text{if } y=0 \\ h(\vec{x},y-1,f_1(\vec{x},q(\vec{x},y),z)) & \text{otherwise.} \end{cases} \end{aligned}$$

f_1 is in $\mathcal{C}$ because $\mathcal{C}$ is closed under primitive recursion, and hence under recursion with substitution of functions in place of parameters (see VIII.8.3.a).

We then define, by α-annihilation, a function $t(\vec{x},y)$ giving the number of steps needed to compute $f(\vec{x},y)$:

$$\begin{aligned} t(\vec{x},0) &= 0 \\ t(\vec{x},y+1) &= 1+t(\vec{x},q(\vec{x},y+1)). \end{aligned}$$

t is in $\mathcal{C}$ because $\mathcal{C}$ is closed under α-annihilation.

We finally prove, by induction on $ord_\alpha(y)$ (the ordinal associated with y in $\prec_\alpha$, i.e. $ord_n(y)$ for the smallest n such that $\alpha < \omega(n)$), that

$$f(\vec{x}, y) = f_1(\vec{x}, y, t(\vec{x}, y)).$$

This shows that f is in $\mathcal{C}$, because $\mathcal{C}$ is closed under composition.

- $ord_\alpha(y) = 0$
  Then $y = 0$ and $t(\vec{x}, 0) = 0$, and

$$f(\vec{x}, 0) = g(\vec{x}) = f_1(\vec{x}, 0, t(\vec{x}, 0)).$$

- $ord_\alpha(y) > 0$
  Then $y > 0$, and by definition

$$\begin{array}{rcl} f(\vec{x}, y) & = & h(\vec{x}, y-1, f(\vec{x}, q(\vec{x}, y))) \\ t(\vec{x}, y) & = & 1 + t(\vec{x}, q(\vec{x}, y)) \\ f_1(\vec{x}, y, t(\vec{x}, y)) & = & h(\vec{x}, y-1, f_1(\vec{x}, q(\vec{x}, y), t(\vec{x}, q(\vec{x}, y)))). \end{array}$$

  But $q(\vec{x}, y) \prec_\alpha y$, i.e. $ord_\alpha(q(\vec{x}, y)) < ord_\alpha(y)$, and by induction hypothesis

$$f(\vec{x}, q(\vec{x}, y)) = f_1(\vec{x}, q(\vec{x}, y), t(\vec{x}, q(\vec{x}, y))),$$

  so that

$$f(\vec{x}, y) = f_1(\vec{x}, y, t(\vec{x}, y)). \quad \square$$

Gödel [1958], Kreisel [1959] and Tait [1959] have extended the notion of primitive recursion from functions to functionals, and have proved that *the ϵ_0-recursive functions are exactly the primitive recursive functionals of type 1*. In particular, there are functions on the integers that can be defined by primitive recursion on higher types, but not by primitive recursion on the integers. See Schwichtenberg [1999] for an exposition of this result.

Functions provably total in Peano Arithmetic ⋆

We have proved in I.3.6 that any consistent formal system $\mathcal{F}$ extending $\mathcal{R}$ *weakly represents* every recursive function, in the sense that for every recursive function f, there is a formula φ such that

$$f(x_1, \ldots, x_n) = y \quad \Leftrightarrow \quad \vdash_{\mathcal{F}} \varphi(\overline{x}_1, \ldots, \overline{x}_n, \overline{y}).$$

Thus, we cannot significantly discriminate among formal systems solely on the basis of the recursive functions representable in them.

However, the fact that a total recursive function f is weakly representable by φ in $\mathcal{F}$ does not mean that $\mathcal{F}$ is able to prove other significant properties relative to φ. For example, we can say that f is *provably total* in $\mathcal{F}$ if

$$\vdash_{\mathcal{F}} (\forall x_1) \cdots (\forall x_n)(\exists y)\varphi(x_1, \ldots, x_n, y).$$

It turns out that it is possible to discriminate among formal systems in a significant way, on the basis of the recursive functions provably total in them.

In particular, while every consistent formal system $\mathcal{F}$ extending $\mathcal{R}$ weakly represents every recursive function, *no consistent formal system $\mathcal{F}$ proves the totality of every total recursive function.* Direct proofs follow by diagonalization or majorization, from the fact that *the recursive functions provably total in a consistent formal system $\mathcal{F}$ form an r.e. class* (Herbrand [1931]).

An indirect proof follows from Gödel's Second Theorem on unprovability of consistency (see p. I.169), once it is noted that *the consistency of a formal system $\mathcal{F}$ is equivalent to the totality of a particular recursive function.* More precisely, let $\{\psi_n\}_{n\in\omega}$ be an enumeration of the sentences of the language of $\mathcal{F}$, and let $R(x, n)$ state that the sequence of formulas coded by x, with the appended two formulas

$$0 = 1 \;\rightarrow\; \psi_n \qquad \text{and} \qquad \psi_n,$$

is a proof of ψ_n in the system $\mathcal{F}$ (i.e. every formula in the sequence is either an axiom of $\mathcal{F}$, or follows from previous formulas by logical rules). Then

$$Con_{\mathcal{F}} \;\Leftrightarrow\; (\forall x)(\exists n)\neg R(x, n),$$

because:

- If $Con_{\mathcal{F}}$, then $0 = 1$ is not provable in $\mathcal{F}$. Let ψ_n be the formula $0 = 1$. Then $R(x, n)$ does not hold for any x, i.e. $(\forall x)\neg R(x, n)$, and hence $(\forall x)(\exists n)\neg R(x, n)$.

- If $\neg Con_{\mathcal{F}}$, then $0 = 1$ is provable in $\mathcal{F}$. Let x be a proof of $0 = 1$ in $\mathcal{F}$. Then $R(x, n)$ holds for every n, i.e. $(\forall n)R(x, n)$, and hence $(\exists x)(\forall n)R(x, n)$.

Notice that, due to the connection between induction and μ-recursion discussed in Section I.1, proving totality of a function defined by μ-recursion amounts to proving induction for the formula used in the μ-recursion. Thus, which recursive functions are provably total in $\mathcal{F}$ strongly depends on the inductive strength of $\mathcal{F}$. More precisely, Kreisel [1952] and Kino [1968] have proved that if $\prec$ is provably well-ordered in $\mathcal{F}$, then every $\prec$-recursive function is provably total in $\mathcal{F}$. The converse holds under quite general conditions, i.e. every recursive function provably total in $\mathcal{F}$ is $\alpha_{\mathcal{F}}$-recursive, where $\alpha_{\mathcal{F}}$ is the l.u.b. of the ordinals provably well-ordered in $\mathcal{F}$. This connects Recursion Theory and Proof Theory, since the ordinal $\alpha_{\mathcal{F}}$ is usually obtained from a consistency proof of $\mathcal{F}$.

The most significant example of the general theory just described is given by $\mathcal{PA}$, where $\alpha_{\mathcal{PA}} = \epsilon_0$. In particular, *the functions provably total in Peano Arithmetic are exactly the* ϵ_0*-recursive functions*, and a similar result also holds for the intuitionistic version of Peano Arithmetic, called Heyting Arithmetic (Kreisel [1952], [1958a]). Friedman [1977] has proved that the same recursive functions are provably total in the classical and intuitionistic versions of a formal system, under quite general conditions.

It is also possible to locate subsystems of Peano Arithmetic corresponding to most of the complexity classes studied in this chapter. For example, *the functions provably total in the system of* Σ_1*-induction are exactly the primitive recursive functions* (Parsons [1970], Mints [1973], Takeuti [1975]), and *the functions provably total in the system of* Δ_0*-induction are exactly the polynomial time computable functions* (Buss [1986]).

For surveys of this topic, see Buchholz and Wainer [1987], Fairtlough and Wainer [1998]. For treatments, see Rose [1984], Buss [1986], Girard [1987], Hájek and Pudlák [1993].

Rate of growth

As for $\mathcal{E}$ and $\mathcal{P}_1$, we obtain a hierarchy for $U(\epsilon_0)$ if we find a sequence of ϵ_0-recursive functions that is cofinal in $U(\epsilon_0)$.

In VIII.8.10, we defined a sequence $\{h_n\}_{n\in\omega}$ of primitive recursive functions cofinal in $\mathcal{P}_1$, and also considered the diagonal function

$$d_{\mathcal{P}_1}(x) = h_x(x).$$

The idea is to consider $d_{\mathcal{P}_1}$ as the natural continuation of the sequence $\{h_n\}_{n\in\omega}$ at ω, and iterate the process in the transfinite.

The next definition extends VIII.8.10, and supplements VIII.7.12 and VIII.8.22.

Definition VIII.9.13 The Fast Growing Hierarchy (Robbin [1965], Constable [1970], Löb and Wainer [1970], [1970a], Schwichtenberg [1971]) *The sequence* $\{h_\alpha\}_\alpha$ *is defined by induction on the countable ordinals, as follows:*

$$\begin{aligned} h_0(x) &= x+1 \\ h_{\alpha+1}(x) &= h_\alpha^{(x)}(x) \\ h_\alpha(x) &= h_{\alpha_x}(x) \quad \textit{if } \alpha \textit{ is a limit} \end{aligned}$$

where, for α a limit, $\{\alpha_x\}_{x\in\omega}$ is an increasing (fundamental) sequence of ordinals with limit α.

In the previous definition we use the fundamental sequences defined in VIII.7.13 when $\alpha \leq \epsilon_0$, and those defined in VIII.8.26 when $\alpha \leq \phi_\omega(0)$.

Since the monotonicity properties of the sequence $\{h_\alpha\}_{\alpha<\epsilon_0}$ are a little more cumbersome to prove than the (quite trivial) ones for $\{h_n\}_{n\in\omega}$ proved in VIII.8.10, we isolate them here.

Proposition VIII.9.14 (Constable [1970], Löb and Wainer [1970], [1970a], Schwichtenberg [1971]) *h_β is dominated by h_α, for any $\beta < \alpha < \epsilon_0$.*

Proof. Before turning to the proof itself, we notice the following properties of the functions h_α, which are proved as in VIII.8.10:

1. $x \leq h_\gamma(x)$
2. $h_\gamma(x) \leq h_{\gamma+1}(x)$, for $x > 0$.

We now turn to the proof, and proceed by induction on α:

- $\alpha = \gamma + 1$
  Since $\beta < \alpha$, either $\beta = \gamma$ or $\beta < \gamma$. The first case is taken care of by Property 2 above. In the second case, by induction hypothesis h_β is dominated by h_γ, and h_γ is dominated by $h_\alpha = h_{\gamma+1}$ by Property 2 above, so h_β is dominated by h_α.

- α *a limit*
  By definition of a fundamental sequence, $\lim_{n\in\omega} \alpha_n = \alpha$. If $\beta < \alpha$, then there is an n such that $\beta < \alpha_n$. By induction hypothesis, h_β is dominated by h_{α_n}, and it is thus enough to prove that, for every n, h_{α_n} is dominated by h_α. This would follow from

  $$(\forall n)(\forall x > 0)(h_{\alpha_n}(x) \leq h_{\alpha_{n+1}}(x)), \qquad \text{(VIII.6)}$$

  since we would then have

  $$x \geq n \geq 1 \ \Rightarrow\ h_{\alpha_n}(x) \leq h_{\alpha_x}(x) \leq h_\alpha(x).$$

We are thus reduced to proving VIII.6, which we do by induction on limit ordinals $\alpha < \epsilon_0$. Recall that, by the Cantor normal form of α, the only two possibilities for a limit ordinal $\alpha < \epsilon_0$ are

$$\alpha = \alpha^* + \omega^{\gamma+1} \qquad \text{or} \qquad \alpha = \alpha^* + \omega^\gamma,\ \gamma \text{ a limit.}$$

We thus have to examine these two cases.

1. $\alpha = \alpha^* + \omega^{\gamma+1}$
   By definition VIII.7.13,
$$\alpha_{n+1} = \alpha^* + \omega^\gamma \cdot (n+1) = \alpha^* + \omega^\gamma \cdot n + \omega^\gamma = \alpha_n + \omega^\gamma.$$
   We thus want to show that
$$(\forall n)(\forall x > 0)(h_{\alpha_n}(x) \leq h_{\alpha_n+\omega^\gamma}(x)),$$
   by induction on γ:
   - $\gamma = 0$
$$\begin{array}{rcll} h_{\alpha_n+\omega^0}(x) & = & h_{\alpha_n+1}(x) & \text{by definition of } \omega^0 \\ & \geq & h_{\alpha_n}(x) & \text{by Property 2 above.} \end{array}$$
   - $\gamma = \theta + 1$
$$\begin{array}{rcll} h_{\alpha_n+\omega^{\theta+1}}(x) & = & h_{(\alpha_n+\omega^{\theta+1})_x}(x) & \text{by definition of } h_{\alpha_n+\omega^{\theta+1}} \\ & = & h_{\alpha_n+\omega^\theta \cdot x}(x) & \text{by definition of } (\alpha_n+\omega^{\theta+1})_x \\ & \geq & h_{\alpha_n}(x) & \text{by induction hypothesis} \\ & & & (x \text{ times}). \end{array}$$
   - γ *a limit*
$$\begin{array}{rcll} h_{\alpha_n+\omega^\gamma}(x) & = & h_{(\alpha_n+\omega^\gamma)_x}(x) & \text{by definition of } h_{\alpha_n+\omega^\gamma} \\ & = & h_{\alpha_n+\omega^{\gamma_x}}(x) & \text{by definition of } (\alpha_n+\omega^\gamma)_x \\ & \geq & h_{\alpha_n}(x) & \text{by induction hypothesis.} \end{array}$$
2. $\alpha = \alpha^* + \omega^\gamma$, γ *a limit*
   Again, by the Cantor normal form of γ, the only two possibilities for a limit ordinal γ are
$$\gamma = \gamma^* + \omega^{\delta+1} \qquad \text{or} \qquad \gamma = \gamma^* + \omega^\delta,\ \delta \text{ a limit.}$$
   We thus have to examine these two subcases.
   (a) $\gamma = \gamma^* + \omega^{\delta+1}$
   By definition VIII.7.13,
$$\alpha_x = \alpha^* + \omega^{\gamma_x},$$
   and
$$\gamma_{n+1} = \gamma^* + \omega^\delta \cdot (n+1) = \gamma^* + \omega^\delta \cdot n + \omega^\delta = \gamma_n + \omega^\delta.$$
   We thus want to show
$$(\forall n)(\forall x > 0)(h_{\alpha^*+\omega^{\gamma_n}}(x) \leq h_{\alpha^*+\omega^{\gamma_n+\omega^\delta}}(x)),$$
   by induction on δ:

- $\delta = 0$

$$\begin{array}{lll} & h_{\alpha^*+\omega^{\gamma_n+\omega^0}}(x) & \\ = & h_{\alpha^*+\omega^{\gamma_n+1}}(x) & \text{by definition of } \omega^0 \\ = & h_{(\alpha^*+\omega^{\gamma_n+1})_x}(x) & \text{by definition of } h_{\alpha^*+\omega^{\gamma_n+1}} \\ = & h_{\alpha^*+\omega^{\gamma_n}\cdot x}(x) & \text{by definition of } (\alpha^*+\omega^{\gamma_n+1})_x \\ \geq & h_{\alpha^*+\omega^{\gamma_n}}(x) & \text{by case 1 } (x>0 \text{ times}). \end{array}$$

- $\delta = \eta + 1$

$$\begin{array}{lll} & h_{\alpha^*+\omega^{\gamma_n+\omega^{\eta+1}}}(x) & \\ = & h_{(\alpha^*+\omega^{\gamma_n+\omega^{\eta+1}})_x}(x) & \text{by definition of } h_{\alpha^*+\omega^{\gamma_n+\omega^{\eta+1}}} \\ = & h_{\alpha^*+\omega^{\gamma_n+\omega^{\eta}\cdot x}}(x) & \text{by definition of } (\alpha^*+\omega^{\gamma_n+\omega^{\eta+1}})_x \\ \geq & h_{\alpha^*+\omega^{\gamma_n}}(x) & \text{by induction hypothesis } (x \text{ times}). \end{array}$$

- *δ a limit*

$$\begin{array}{lll} & h_{\alpha^*+\omega^{\gamma_n+\omega^{\delta}}} & \\ = & h_{(\alpha^*+\omega^{\gamma_n+\omega^{\delta}})_x}(x) & \text{by definition of } h_{\alpha^*+\omega^{\gamma_n+\omega^{\delta}}} \\ = & h_{\alpha^*+\omega^{\gamma_n+\omega^{\delta_x}}}(x) & \text{by definition of } (\alpha^*+\omega^{\gamma_n+\omega^{\delta}})_x \\ \geq & h_{\alpha^*+\omega^{\gamma_n}}(x) & \text{by induction hypothesis.} \end{array}$$

(b) *$\gamma = \gamma^* + \omega^\delta$, δ a limit*
Again, by the Cantor normal form of δ, the only possibilities for a limit ordinal δ are

$$\delta = \delta^* + \omega^{\eta+1} \qquad \text{or} \qquad \delta = \delta^* + \omega^\eta,\ \eta \text{ a limit.}$$

We thus have to examine these two subcases, the first of which is proved by induction on η (using subcase 2.(a) as the basis), and the second of which produces two further possibilities, and so on.

It is clear that the process above has to finish after a finite number of steps, since the ordinals we consider (in particular α) are less than ϵ_0, and their Cantor normal forms use only smaller exponents. In particular, after a finite number of steps, the second possibility no longer arises and the induction ends. $\Box$

The proof above of

$$(\forall n)(\forall x > 0)(h_{\alpha_n}(x) \leq h_{\alpha_{n+1}}(x))$$

uses in an essential way the uniformity properties of the system of notations for $\alpha < \epsilon_0$ introduced in VIII.7.13. Schmidt [1976] has shown that for every countable ordinal α there are systems of notations with the required uniformity properties.

We can now state the analogue of VIII.8.10.

Theorem VIII.9.15 A Skeleton for $U(\epsilon_0)$ (Wainer [1970], Schwichtenberg [1971])

1. *h_α is ϵ_0-recursive, for each $\alpha < \epsilon_0$*
2. *every ϵ_0-recursive function is dominated by some h_α, in the sense that given f ϵ_0-recursive, there is an $\alpha < \epsilon_0$ such that $f(\vec{x}) \leq h_\alpha(\sum \vec{x})$, for almost every $\vec{x}$*
3. *the diagonal function $d_{U(\epsilon_0)} = h_{\epsilon_0}$ dominates every ϵ_0-recursive function, and thus it is not ϵ_0-recursive.*

Proof. Part 3 follows from Part 2 and VIII.9.14, as follows. If $\alpha < \epsilon_0$, then there is an n such that $\alpha \leq (\epsilon_0)_n$, and so for almost every $x \geq n$

$$\begin{array}{rcll} h_\alpha(x) & \leq & h_{(\epsilon_0)_n}(x) & \text{by VIII.9.14, since } \alpha \leq (\epsilon_0)_n \\ & \leq & h_{(\epsilon_0)_x}(x) & \text{by VIII.9.14, since } n \leq x \\ & = & h_{\epsilon_0}(x) & \text{by definition of } h_{\epsilon_0}. \end{array}$$

Part 1 is proved by showing, by induction on n, that $\{h_\alpha\}_{\alpha<\omega(n)}$ can be defined by ϵ_0-recursion. As in VIII.8.11, the sequence can be described by the following equations, the first three defining h_α by recursion on α, and the second two defining $h_\alpha^{(z)}$ by recursion on z:

$$\begin{array}{rcl} h_0(x) & = & x+1 \\ h_{\alpha+1}(x) & = & h_\alpha^{(x)}(x) \\ h_\alpha(x) & = & h_{\alpha_x}(x), \text{ if } \alpha \text{ is a limit} \\ h_\alpha^{(0)}(x) & = & x \\ h_\alpha^{(z+1)}(x) & = & h_\alpha(h_\alpha^{(z)}(x)). \end{array}$$

We now have to replace the recursion on ordinals α by a recursion on numbers.

The first idea that comes to mind is that, since we restrict our attention to $\alpha < \omega(n)$ for a fixed n, we can replace α by $num_n(\alpha)$ (defined in VIII.9.5). There is one problem, in doing this: the equations above define a function by nested recursion on *two* variables (α and z), while the definition of nested ordinal recursion (VIII.9.3) allows for nestings on only *one* variable. But we only want to show that the sequence $\{h_\alpha\}_{\alpha<\omega(n)}$ is ϵ_0-recursive, not necessarily $\omega(n)$-recursive, and we can thus use an ordinal larger than $\omega(n)$ for the recursion (exactly as the function defined in VIII.8.11 by nested recursion on two variables over ω can be defined by nested recursion on a single variable over ω^2, by VIII.9.7).

The second idea is to separate the two recursions. First, we define each h_α by means of the first three equations, using a recursion on $\omega(n)$. Second, we define the iterations $h_\alpha^{(z)}$ by means of the last two equations, using a further recursion on ω, for each α. This can be done by defining a single function

$$h_1(\omega^\alpha + z, x) = h_\alpha^{(z)}(x)$$

by nested recursion on a single variable, as follows (thinking of h_α as $h_\alpha^{(1)}$):

$$\begin{array}{rcl} h_1(\omega^0 + 1, x) & = & x + 1 \\ h_1(\omega^{\alpha+1} + 1, x) & = & h_1(\omega^\alpha + x, x) \\ h_1(\omega^\alpha + 1, x) & = & h_1(\omega^{\alpha_x} + 1, x), \text{ if } \alpha \text{ is a limit} \\ h_1(\omega^\alpha, x) & = & x \\ h_1(\omega^\alpha + z + 1, x) & = & h_1(\omega^\alpha + 1, h_1(\omega^\alpha + z, x)). \end{array}$$

We can now indeed substitute numbers $num_n(\alpha)$ for ordinals $\alpha < \omega(n)$, and hence $num_{n+1}(\beta)$ for ordinals $\beta < \omega(n+1)$. By looking at the details of the definitions of canonical well-orderings (VIII.9.5) and fundamental sequences (VIII.7.13) up to ϵ_0, it is easy to check that the following predicates and functions are primitive recursive:

$$\begin{array}{lcll} P_0(y) & \Leftrightarrow & ord_{n+1}(y) = \omega^0 + 1 & \\ P_1(y) & \Leftrightarrow & ord_{n+1}(y) = \omega^{\alpha+1} + 1 & \text{for some } \alpha < \omega(n) \\ P_2(y) & \Leftrightarrow & ord_{n+1}(y) = \omega^\alpha + 1 & \text{for some limit } \alpha < \omega(n) \\ P_3(y) & \Leftrightarrow & ord_{n+1}(y) = \omega^\alpha & \text{for some } \alpha < \omega(n) \\ P_4(y) & \Leftrightarrow & ord_{n+1}(y) = \omega^\alpha + z + 1 & \text{for some } \alpha < \omega(n) \text{ and } z \end{array}$$

and

$$\begin{array}{rcl} g_1(y,x) & = & \begin{cases} num_{n+1}(\omega^\alpha + x) & \text{if } ord_{n+1}(y) = \omega^{\alpha+1} + 1 \\ 0 & \text{otherwise} \end{cases} \\ g_2(y,x) & = & \begin{cases} num_{n+1}(\omega^{\alpha_x} + 1) & \text{if } ord_{n+1}(y) = \omega^\alpha + 1 \text{ and } \alpha \text{ is a limit} \\ 0 & \text{otherwise} \end{cases} \\ g_3(y) & = & \begin{cases} num_{n+1}(\omega^\alpha + 1) & \text{if } ord_{n+1}(y) = \omega^\alpha + z + 1 \\ 0 & \text{otherwise} \end{cases} \\ g_4(y) & = & \begin{cases} num_{n+1}(\omega^\alpha + z) & \text{if } ord_{n+1}(y) = \omega^\alpha + z + 1 \\ 0 & \text{otherwise.} \end{cases} \end{array}$$

For example, $P_2(y)$ can be defined by noting that:

- $ord_n(y) = 0$ if and only if $y = 0$
- $ord_n(y)$ is a successor if and only if the exponent of p_0 in the decomposition of $y + 1$ is not 0
- $ord_n(y)$ is a limit if and only if $y \neq 0$ and $ord_n(y)$ is not a successor

- $ord_{n+1}(y) = \omega^\alpha + 1$ for some limit $\alpha < \omega(n)$ if and only if $y+1 = p_0 \cdot p_a$ for some a such that $ord_n(a)$ is a limit.

The other predicates and functions are defined similarly.

We can now define a function h_2 by nested $\prec_{n+1}$-recursion, as follows:

$$\begin{array}{rcl} h_2(0,x) & = & 0 \\ h_2(y+1,x) & = & \begin{cases} x+1 & \text{if } P_0(y+1) \\ h_2(g_1(y+1,x),x) & \text{if } P_1(y+1) \\ h_2(g_2(y+1,x,),x) & \text{if } P_3(y+1) \\ x & \text{if } P_3(y+1) \\ h_2(g_3(y+1),h_2(g_4(y+1),x)) & \text{if } P_4(y+1) \\ 0 & \text{otherwise.} \end{cases} \end{array}$$

This is indeed a $\prec_{n+1}$-recursion, because obviously $g_i(y+1,x)$ $(i=1,2)$ and $g_j(y+1)$ $(j=3,4,5)$ precede $y+1$ in the ordering $\prec_{n+1}$ by definition (being notations for smaller ordinals), and the known functions are all primitive recursive. Moreover, for every $\alpha < \omega(n)$,

$$h_2(num_{n+1}(\omega^\alpha + z), x) = h_1(\omega^\alpha + z, x) = h_\alpha^{(z)}(x).$$

This completes the proof of Part 1.

Part 2 is proved by induction on the characterization of ϵ_0-recursive functions given by VIII.9.12, as follows:

- *initial functions*
  They are all dominated by h_0.

- *primitive recursive operations*
  They are treated as in VIII.8.10, using the monotonicity property proved in VIII.9.14. In particular, if h_α dominates a finite number of functions, then $h_{\alpha+\omega}$ dominates every function built from them by primitive recursive operations.

- $\omega(n)$*-annihilation*
  If f has been defined by $\omega(n)$-annihilation from q, i.e.

  $$\begin{array}{rcl} f(\vec{x},0) & = & 0 \\ f(\vec{x},y+1) & = & 1 + f(\vec{x}, q(\vec{x},y+1)) \end{array}$$

  where
  $$q(\vec{x},0) = 0 \qquad \text{and} \qquad q(\vec{x},y+1) \prec_n y+1,$$

  by induction hypothesis, there is an $\alpha < \epsilon_0$ such that, for almost every $\vec{x}$ and y, $q(\vec{x},y) \le h_\alpha(y + \sum \vec{x})$.

We show that there is a function $t(\vec{x}, y)$ elementary in q such that, for almost every $\vec{x}$ and y,

$$f(\vec{x}, y) \leq h_{\alpha+ord_n(y)}(t(\vec{x}, y)). \tag{VIII.7}$$

This will not be enough, however, since the function on the right hand side depends on y, via $ord_n(y)$. But by using certain uniformities, we can show that, for almost every $\vec{x}$,

$$f(\vec{x}, y) \leq h_{\alpha+\omega(n)}(t(\vec{x}, y)). \tag{VIII.8}$$

Then we are done. Since t is elementary in q and q is dominated by h_α, t is dominated by $h_{\alpha+1}$ by VIII.7.8 and, for almost every $\vec{x}$ and y,

$$\begin{array}{rcll}
f(\vec{x}, y) & \leq & h_{\alpha+\omega(n)}(h_{\alpha+1}(y + \sum \vec{x})) & \text{by the proof of VIII.9.14} \\
& \leq & h_{\alpha+\omega(n)}(h_{\alpha+\omega(n)}(y + \sum \vec{x})) & \text{by VIII.9.14} \\
& = & h^{(2)}_{\alpha+\omega(n)}(y + \sum \vec{x}) & \\
& \leq & h^{(y+\sum \vec{x})}_{\alpha+\omega(n)}(y + \sum \vec{x}) & \text{if } 2 \leq y + \sum \vec{x} \\
& \leq & h_{\alpha+\omega(n)+1}(y + \sum \vec{x}). & \text{by definition of } h_{\alpha+\omega(n)+1}.
\end{array}$$

To finish the proof, we have to find a function t satisfying VIII.7. We proceed by induction on $ord_n(y)$:

- $ord_n(y) = 0$
  Then $y = 0$ and $f(\vec{x}, 0) = 0$, so there is nothing to do.

- $ord_n(y) > 0$
  Then

$$\begin{array}{rcll}
f(\vec{x}, y) & = & 1 + f(\vec{x}, q(\vec{x}, y)) & \text{by definition of } f \\
& \leq & 1 + h_{\alpha+ord_n(q(\vec{x},y))}(t(\vec{x}, q(\vec{x}, y))) & \text{by induction hypothesis.}
\end{array}$$

If we define t in such a way that there is a z such that, for every $\vec{x}$ and y,

$$t(\vec{x}, q(\vec{x}, y)) \leq h^{(z)}_\alpha(t(\vec{x}, y)) \tag{VIII.9}$$

then we can resume the sequence of inequalities, as follows:

$$\begin{array}{cll}
\leq & 1 + h_{\alpha+ord_n(q(\vec{x},y))}(h^{(z)}_\alpha(t(\vec{x}, y))) & \text{by the proof of VIII.9.14} \\
= & h_0(h_{\alpha+ord_n(q(\vec{x},y))}(h^{(z)}_\alpha(t(\vec{x}, y)))) & \text{by definition of } h_0 \\
\leq & h^{(z+2)}_{\alpha+ord_n(q(\vec{x},y))}(t(\vec{x}, y)) & \text{by VIII.9.14} \\
\leq & h^{(t(\vec{x},y))}_{\alpha+ord_n(q(\vec{x},y))}(t(\vec{x}, y)) & \text{if } z + 2 \leq t(\vec{x}, y) \\
= & h_{\alpha+ord_n(q(\vec{x},y))+1}(t(\vec{x}, y)) & \text{by definition of } h_{\alpha+ord_n q(\vec{x},y)+1}.
\end{array}$$

Now $q(\vec{x}, y) \prec_n y$, since $y > 0$, and thus $ord_n(q(\vec{x}, y)) + 1 \leq ord_n(y)$. If we prove that

$$\beta \leq \gamma < \omega(n) \text{ and } num_n(\beta) \leq a \quad \Rightarrow \quad h_{\alpha+\beta}(a) \leq h_{\alpha+\gamma}(a) \quad \text{(VIII.10)}$$

and define t in such a way that

$$num_n[ord_n(q(\vec{x}, y)) + 1] \leq t(\vec{x}, y), \qquad \text{(VIII.11)}$$

then using VIII.10 with

$$\beta = ord_n(q(\vec{x}, y)) + 1 \qquad \gamma = ord_n(y) \qquad a = t(\vec{x}, y),$$

we can conclude the sequence of inequalities, as follows:

$$f(\vec{x}, y) \leq h_{\alpha+ord_n(y)}(t(\vec{x}, y)),$$

and VIII.7 is proved.

We are thus left with the three properties VIII.9, VIII.10 and VIII.11 to prove. We satisfy VIII.9 and VIII.11 by appropriately defining t and z, and verify VIII.10 independently.

We start with VIII.9 and VIII.11, i.e. with the definition of t and z. By definition of canonical well-orderings (VIII.9.5), adding one to an ordinal means multiplying its notation by $p_0 = 2$. Now $ord_n(q(\vec{x}, y))$ is coded by $q(\vec{x}, y) + 1$ (by definition VIII.9.5), so $ord_n(q(\vec{x}, y)) + 1$ is coded by $2(q(\vec{x}, y) + 1)$. To satisfy VIII.11 and also have $t(\vec{x}, y) \geq z + 2$ (used above), it is enough to let

$$t(\vec{x}, y) = z + 2 \cdot q(\vec{x}, y) + 2 + y + \sum \vec{x}$$

(the reason for also adding $y + \sum \vec{x}$ will be apparent in a moment). We now have to determine z in such a way that VIII.9 holds. But

$$t(\vec{x}, q(\vec{x}, y)) = z + 2 \cdot q(\vec{x}, q(\vec{x}, y)) + 2 + q(\vec{x}, y) + \sum \vec{x},$$

and thus $t(\vec{x}, q(\vec{x}, y)) - z$ is elementary in q (and independent of z). Since q is dominated by h_α almost everywhere, there is a z such that, for every $\vec{x}$ and y,

$$\begin{aligned} t(\vec{x}, q(\vec{x}, y)) &\leq z + h_\alpha^{(z)}(y + \textstyle\sum \vec{x}) \\ &\leq h_\alpha^{(z)}(z + y + \textstyle\sum \vec{x}) \\ &\leq h_\alpha^{(z)}(t(\vec{x}, y)) \end{aligned}$$

by VIII.7.21, since $z + y + \sum \vec{x} \leq t(\vec{x}, y)$, and thus VIII.9 is also satisfied.

This disposes of everything except VIII.10, i.e.

$$\beta < \gamma < \omega(n) \quad \text{and} \quad num_n(\beta) \leq x \;\Rightarrow\; h_{\alpha+\beta}(x) \leq h_{\alpha+\gamma}(x),$$

which we now prove by induction on $\gamma < \omega(n)$ (we concentrate on $\beta < \gamma$ because the case $\beta = \gamma$ is trivial).

- $\gamma = 0$
  There is nothing to prove.

- $\gamma = \delta + 1$
  If $\beta < \gamma$, either $\beta = \delta$ or $\beta < \delta$. The first case is trivial, and in the second case
  $$\begin{array}{rcll} h_{\alpha+\beta}(x) & \leq & h_{\alpha+\delta}(x) & \text{by induction hypothesis} \\ & \leq & h_{\alpha+\delta+1}(x) & \text{by the proof of VIII.9.14} \\ & = & h_{\alpha+\gamma}(x) & \text{by case hypothesis.} \end{array}$$

- γ *is a limit*
  If we knew that $\beta \leq \gamma_x$ follows from the hypotheses, then we would have
  $$\begin{array}{rcll} h_{\alpha+\beta}(x) & \leq & h_{\alpha+\gamma_x}(x) & \text{by VIII.9.14, since } \beta \leq \gamma_x \\ & = & h_{(\alpha+\gamma)_x}(x) & \text{by definition of } (\alpha+\gamma)_x \\ & = & h_{\alpha+\gamma}(x) & \text{by definition of } h_{\alpha+\gamma}. \end{array}$$

We now prove by induction on n that if $x \neq 0$, then

$$\beta < \gamma < \omega(n) \quad \text{and} \quad num_n(\beta) \leq x \quad \text{and} \quad \gamma \text{ is a limit } \Rightarrow \beta < \gamma_x,$$

(the conclusion $<$ is stronger than the required $\leq$, but will be needed for the induction to work).

For $n = 0$, there is nothing to prove. We thus concentrate on the case of $n+1$. Let $\gamma < \omega(n+1)$ be a limit, and $\beta < \gamma$. In their Cantor normal forms, β and γ will have a common beginning δ, and there will be a first place where they differ. We can thus suppose

$$\beta = \delta + \omega^{\sigma} \cdot b + \omega^{\sigma'} \cdot b' + \cdots \qquad \text{and} \qquad \gamma = \delta + \omega^{\tau} \cdot c + \cdots,$$

where $\sigma > \sigma'$, and

$$\sigma < \tau \qquad \text{or} \qquad \sigma = \tau \ \wedge \ b < c.$$

By definition VIII.7.13, we have

$$\gamma_x \geq \delta + \omega^{\tau} \cdot (c-1) + (\omega^{\tau})_x,$$

where the precise form of $(\omega^{\tau})_x$ depends on whether τ is a successor or a limit. We thus have to examine these two cases.

1. $\sigma < \tau$
   Let $num_{n+1}(\beta) \leq x$. By definition VIII.9.5, $p^{b}_{num_n(\sigma)} - 1 \leq num_{n+1}(\beta)$. Thus $x > b$ if $x \neq 0$, and $num_n(\sigma) \leq x$ if $b \neq 0$. We have two cases:

(a) τ *a successor*

$$\begin{array}{rlll} \gamma_x & \geq & \delta + \omega^\tau \cdot (c-1) + (\omega^\tau)_x & \\ & = & \delta + \omega^\tau \cdot (c-1) + \omega^{\tau-1} \cdot x & \text{by definition of } (\omega^\tau)_x \\ & \geq & \delta + \omega^\tau \cdot (c-1) + \omega^\sigma \cdot x & \text{because } \tau > \sigma \\ & \geq & \delta + \omega^\tau \cdot (c-1) + \omega^\sigma \cdot (b+1) & \text{because } x > b \\ & > & \beta & \text{because } \beta = \delta + \omega^\sigma \cdot b \cdots \end{array}$$

(b) τ *a limit*

$$\begin{array}{rlll} \gamma_x & \geq & \delta + \omega^\tau \cdot (c-1) + (\omega^\tau)_x & \\ & = & \delta + \omega^\tau \cdot (c-1) + \omega^{\tau_x} & \text{by definition of } (\omega^\tau)_x. \end{array}$$

If $\beta = \delta$, then we are finished.

If $\beta > \delta$, then $b \neq 0$, and so $num_n(\sigma) \leq x$ as noted above. Thus

$$\sigma < \tau \qquad \text{and} \qquad num_n(\sigma) \leq x \qquad \text{and} \qquad \gamma \text{ is a limit}$$

by the case hypotheses, so $\sigma < \tau_x$ by the induction hypothesis and

$$\delta + \omega^\tau \cdot (c-1) + \omega^{\tau_x} > \beta.$$

2. $\sigma = \tau \ \wedge \ b < c$
Let $num_{n+1}(\beta) \leq x$. By definition VIII.9.5, $p^{b'}_{num_n(\sigma')} - 1 \leq num_{n+1}(\beta)$. Thus $x > b'$ if $x \neq 0$, and $num_n(\sigma') \leq x$ if $b' \neq 0$. We have two cases:

(a) τ *a successor*

$$\begin{array}{rlll} \gamma_x & \geq & \delta + \omega^\tau \cdot (c-1) + (\omega^\tau)_x & \\ & = & \delta + \omega^\tau \cdot (c-1) + \omega^{\tau-1} \cdot x & \text{by definition of } (\omega^\tau)_x \\ & = & \delta + \omega^\sigma \cdot (c-1) + \omega^{\sigma-1} \cdot x & \text{because } \sigma = \tau \\ & \geq & \delta + \omega^\sigma \cdot b + \omega^{\sigma-1} \cdot x & \text{because } c > b \\ & \geq & \delta + \omega^\sigma \cdot b + \omega^{\sigma'} \cdot x & \text{because } \sigma > \sigma' \\ & > & \beta & \text{because } x > b'. \end{array}$$

(b) τ *a limit*

$$\begin{array}{rlll} \gamma_x & \geq & \delta + \omega^\tau \cdot (c-1) + (\omega^\tau)_x & \\ & = & \delta + \omega^\tau \cdot (c-1) + \omega^{\tau_x} & \text{by definition of } (\omega^\tau)_x \\ & = & \delta + \omega^\sigma \cdot (c-1) + \omega^{\sigma_x} & \text{because } \sigma = \tau \\ & \geq & \delta + \omega^\sigma \cdot b + \omega^{\sigma_x} & \text{because } c > b. \end{array}$$

If $\beta = \delta + \omega^\sigma \cdot b$, then we are finished.

If $\beta > \delta + \omega^\sigma \cdot b$, then $b' \neq 0$, and so $num_n(\sigma') \leq x$ as noted above. Thus

$$\sigma' < \sigma \quad \text{and} \quad num_n(\sigma') \leq x \quad \text{and} \quad \sigma \text{ is a limit}$$

by the case hypotheses, so $\sigma' < \sigma_x$ by the induction hypothesis and

$$\delta + \omega^\sigma \cdot b + \omega^{\sigma_x} > \beta.$$

Notice that this is the place where we need the stronger conclusion for the induction to work.

We have thus succeeded in proving Equation VIII.7, i.e. in defining t such that, for almost every $\vec{x}$ and y,

$$f(\vec{x}, y) \leq h_{\alpha + ord_n(y)}(t(\vec{x}, y)).$$

If we now show that $ord_n(y) \leq (\omega(n))_y$, then from

$$num_n(ord_n(y)) = y \leq t(\vec{x}, y)$$

(which holds by definition of t) we have

$$\begin{array}{rcll} f(\vec{x}, y) & \leq & h_{\alpha + ord_n(y)}(t(\vec{x}, y)) & \text{by VIII.7} \\ & \leq & h_{\alpha + (\omega(n))_y}(t(\vec{x}, y)) & \text{by VIII.9.14, since } ord_n(y) \leq (\omega(n))_y \\ & \leq & h_{\alpha + (\omega(n))_{t(\vec{x},y)}}(t(\vec{x}, y)) & \text{by VIII.9.14, since } y \leq t(\vec{x}, y) \\ & = & h_{\alpha + \omega(n)}(t(\vec{x}, y)) & \text{by definition of } h_{\alpha + \omega(n)} \end{array}$$

for almost every $\vec{x}$ and y, and this proves VIII.8.

We thus show $ord_n(y) \leq (\omega(n))_y$, for every $n \geq 1$ and y, by induction on n.

- $n = 0$
  Since $ord_1(y) = y$, $\omega(1) = \omega$ and $(\omega(1))_y = y$, there is nothing to prove.

- $n + 1$
  We proceed by induction on y.

  - $y = 0$
    Since $ord_{n+1}(0) = 0$, there is nothing to prove.
  - $y > 0$
    Let $y = p_{b_m}^{a_m} \cdots p_{b_1}^{a_1} - 1$, with $a_1 > 0$ and $b_1 > \cdots > b_m$.
    Since $b_1 < y$, by induction hypothesis on y we have

$$ord_n(b_1) \leq (\omega(n))_{b_1} < (\omega(n))_y.$$

Since, by definition,

$$ord_{n+1}(y) = \omega^{ord_n(b_1)} \cdot a_1 + \cdots + \omega^{ord_n(b_m)} \cdot a_m,$$

and

$$ord_n(b_m) < \cdots < ord_n(b_1) < (\omega(n))_y,$$

we then have

$$ord_{n+1}(y) < (\omega(n+1))_y.$$

The proof of Part 2 is thus concluded. $\Box$

Hierarchies

By using VIII.9.15, we can immediately obtain hierarchies for the ϵ_0-recursive functions, in the same way as we did for the primitive recursive functions in VIII.8.13. Recall that $\mathcal{E}(g)$ is the class of functions elementary in g (VIII.7.16).

Definition VIII.9.16 The Extended Grzegorczyk Hierarchy (Constable [1970], Löb and Wainer [1970], [1970a], Wainer [1970], Schwichtenberg [1971]) *For* $3 \leq \alpha \leq \epsilon_0$*,*

1. *if* $\alpha = \beta + 1$*, then* $\boldsymbol{\mathcal{E}_\alpha} = \mathcal{E}(h_\beta)$
2. *if* α *is a limit, then* $\boldsymbol{\mathcal{E}_\alpha} = \bigcup_{3 \leq \beta < \alpha} \mathcal{E}_\beta$*.*

For $3 \leq \alpha < \omega$, the classes $\mathcal{E}_\alpha$ coincide with those defined in VIII.8.12. Moreover, $\mathcal{E}_\omega = \mathcal{P}_1 = U(\omega)$, by VIII.8.13.2.

The next result is the analogue of VIII.8.13.

Theorem VIII.9.17 Hierarchy Theorem for $U(\epsilon_0)$ (Constable [1970], Löb and Wainer [1970], [1970a], Wainer [1970], Schwichtenberg [1971])

1. $\mathcal{E}_\alpha \subset \mathcal{E}_\beta$*, for each* α *and* β *such that* $3 \leq \alpha < \beta \leq \epsilon_0$
2. $U(\epsilon_0) = \mathcal{E}_{\epsilon_0}$*.*

Proof. Part 1 follows from the following two facts, for $2 \leq \alpha < \beta < \epsilon_0$:

- h_α *is elementary in* h_β *(so* $\mathcal{E}_{\alpha+1} \subseteq \mathcal{E}_{\beta+1}$*)*
  Since $\beta < \epsilon_0$, there is an n such that $\beta \leq \omega(n)$. The proof of VIII.9.15.1 shows that there is a recursive predicate R such that

$$h_\alpha(x) = \mu y\, R(num_n(\alpha), x, y),$$

namely

$$R(num_n(\alpha), x, y) \;\Leftrightarrow\; h_2(num_{n+1}(\omega^\alpha + 1), x) = y.$$

Then h_α is μ-recursive uniformly in $\alpha < \omega(n)$.

Since $\alpha < \beta$, by VIII.9.14 h_α is dominated almost everywhere by h_β, and hence everywhere by a finite modification of h_β, i.e. by a function elementary in h_β. By closure of $\mathcal{E}$ under bounded μ-recursion (VIII.7.3.1), h_α is then elementary in h_β.

- *h_β is not elementary in h_α (so $\mathcal{E}_{\alpha+1} \neq \mathcal{E}_{\beta+1}$)*
  If $\alpha < \beta$, then h_β dominates $h_{\alpha+1}$ (by VIII.9.14), and hence every function elementary in h_α (by VIII.7.21.b). In particular, h_β is not elementary in h_α.

With these two facts, we can prove $\mathcal{E}_\alpha \subset \mathcal{E}_{\alpha+1}$ by induction on α such that $3 \leq \alpha < \epsilon_0$:

- $\alpha = 3$
  We already know that $\mathcal{E}_3 \subset \mathcal{E}_4$ from VIII.8.13.

- *α a successor*
  This follows from the two properties proved above.

- *α a limit*
  By definition, $\mathcal{E}_\alpha = \bigcup_{3 \leq \gamma < \alpha} \mathcal{E}_\gamma$.

  By the first property above, if $3 \leq \gamma < \alpha$, then $\mathcal{E}_\gamma \subseteq \mathcal{E}_{\gamma+1} \subseteq \mathcal{E}_{\alpha+1}$, and so $\bigcup_{3 \leq \gamma < \alpha} \mathcal{E}_\gamma \subseteq \mathcal{E}_{\alpha+1}$.

  By the second property above, if $3 \leq \gamma < \alpha$, then $h_\alpha \notin \mathcal{E}_{\gamma+1}$, hence $h_\alpha \notin \mathcal{E}_\gamma$, and so $h_\alpha \notin \bigcup_{3 \leq \gamma < \alpha} \mathcal{E}_\gamma$.

Part 1 now follows by induction, since at limit stages we take unions.

To prove Part 2, first notice that $\mathcal{E}_{\epsilon_0} \subseteq U(\epsilon_0)$ because $U(\epsilon_0)$ is closed under (primitive recursive, and hence) elementary operations by definition and, for each $\alpha < \epsilon_0$, h_α is ϵ_0-recursive by VIII.9.15.1. Conversely, $U(\epsilon_0) \subseteq \mathcal{E}_{\epsilon_0}$ follows by induction on the characterization VIII.9.12 of $U(\epsilon_0)$:

- *initial functions*
  They are all elementary, and hence in $\mathcal{E}_3$.

- *primitive recursive operations*
  The proof of VIII.8.13.2 shows that any function defined by primitive recursive operations from functions in $\mathcal{E}_\alpha$ is in $\mathcal{E}_{\alpha+\omega}$. If $\alpha < \epsilon_0$, then $\alpha + \omega < \epsilon_0$. Thus $\mathcal{E}_{\epsilon_0}$ is closed under primitive recursive operations.

- *α-annihilation*
  If $f \in U(\epsilon_0)$ has been defined from q by α-annihilation for $\alpha < \epsilon_0$, i.e.

$$\begin{array}{rcl} f(\vec{x}, 0) & = & 0 \\ f(\vec{x}, y+1) & = & 1 + f(\vec{x}, q(\vec{x}, y+1)) \end{array}$$

  where

$$q(\vec{x}, 0) = 0 \qquad \text{and} \qquad q(\vec{x}, y+1) \prec_\alpha y+1,$$

  by induction hypothesis $q \in \mathcal{E}_{\epsilon_0}$. We first define, by primitive recursion, a function $f_1(\vec{x}, y, z)$ giving the result of z applications of q, starting from y:

$$\begin{array}{rcl} f_1(\vec{x}, y, 0) & = & y \\ f_1(\vec{x}, y, z+1) & = & q(\vec{x}, f_1(\vec{x}, y, z)). \end{array}$$

  $f_1 \in \mathcal{E}_{\epsilon_0}$ because $\mathcal{E}_{\epsilon_0}$ is closed under primitive recursion. Moreover,

$$f(\vec{x}, y) = \mu z. \, (f_1(\vec{x}, y, z) = 0).$$

  Since f is ϵ_0-recursive, by VIII.9.15.2 f is dominated almost everywhere by h_α, for some $\alpha < \epsilon_0$, and hence everywhere by a finite modification of h_α, i.e. by a function g in $\mathcal{E}_{\epsilon_0}$. Thus

$$f(\vec{x}, y) = \mu z_{\leq g(\vec{x}, y)} \, (f_1(\vec{x}, y, z) = 0),$$

  and f is defined by bounded μ-recursion from functions in $\mathcal{E}_{\epsilon_0}$. Then also f is in $\mathcal{E}_{\epsilon_0}$, because $\mathcal{E}_{\epsilon_0}$ is closed under (elementary operations, and hence under) bounded μ-recursion. $\Box$

Various refinements of the previous result have been obtained, in particular:

- $\mathcal{E}_{\omega^n} = \mathcal{P}_n$, for any $n \geq 1$ (Robbin [1965], Löb and Wainer [1970a])
- $\mathcal{E}_{\alpha \cdot \omega} = U(\omega^\alpha) = N(\omega \cdot \alpha)$, for $0 \leq \alpha < \epsilon_0$ (Wainer [1972]).

The classes of the Extended Grzegorczyk Hierarchy have the same computational properties as the classes of the Grzegorczyk Hierarchy, and the next result is the analogue of VIII.7.6, VIII.8.8 and VIII.8.14.

Theorem VIII.9.18 Ritchie-Cobham Property for $\boldsymbol{\mathcal{E}_\alpha}$ (Constable [1970], Wainer [1970]) *For any α such that $3 \leq \alpha \leq \epsilon_0$, the following are equivalent:*

1. *f belongs to $\mathcal{E}_\alpha$*

2. *f is computable in time belonging to $\mathcal{E}_\alpha$*
3. *f is computable in space belonging to $\mathcal{E}_\alpha$.*

Proof. Since we take unions at limit stages, it is enough to prove the result for successor ordinals $\alpha + 1$. Since, for $2 \leq \alpha < \epsilon_0$, $\mathcal{E}_{\alpha+1}$ is the class of functions elementary in h_α, by VIII.7.18 it is enough to prove the following:

- *h_α is computable in time elementary in h_α, for any $\alpha < \epsilon_0$*
  Since

$$\begin{array}{rcl} h_0(x) & = & x+1 \\ h_{\alpha+1}(x) & = & h_\alpha^{(x)}(x) \\ h_\alpha(x) & = & h_{\alpha_x}(x), \text{ if } \alpha \text{ is a limit} \\ h_\alpha^{(0)}(x) & = & x \\ h_\alpha^{(z+1)}(x) & = & h_\alpha(h_\alpha^{(z)}(x)), \end{array}$$

  if $T_h(\alpha, z, x)$ is the time needed to compute $h_\alpha^{(z)}(x)$, and if we suppose that no time is needed to compute the identity function (used in the definition for $z = 0$), and a single step is needed to compute the successor function (used in the definition for $\alpha = 0$), then roughly (as in VIII.8.14):

$$\begin{array}{rcl} T_h(0,1,x) & = & 1 \\ T_h(\alpha+1,1,x) & = & T_h(\alpha,x,x) \\ T_h(\alpha,1,x) & = & T_h(\alpha_x,1,x), \text{ if } \alpha \text{ is a limit} \\ T_h(\alpha,0,x) & = & 0 \\ T_h(\alpha,z+1,x) & = & \sum_{y \leq z} T_h(\alpha,1,h_\alpha^{(y)}(x)). \end{array}$$

  Since $\alpha < \epsilon_0$, there is an n such that $\alpha < \omega(n)$. We first show, by induction on $\alpha < \omega(n)$, that $T_h(\alpha, z, x)$ is bounded by a function elementary in $h_{\alpha+1}$, uniformly in α.

  - $\alpha = 0$
    $T_h(0, z, x)$ is the time needed to compute $h_0^{(z)}(x)$, and is elementary by the proof of VIII.8.14.
  - *α a successor*
    The claim holds either trivially (for $z = 0$), or by induction hypothesis (for $z = 1$), or as in the proof of VIII.8.14 (for $z + 1$).
  - *α a limit*
    The claim holds by induction hypothesis, using uniformity and the

fact that the system of notations for ordinals less than $\omega(n)$ is elementary (since, by definition VIII.9.5, notations for ordinals less than $\omega(n)$ require only n iterations of the exponential function).

It is now immediate to show, by induction on $\alpha < \omega(n)$, that $T_h(\alpha, 1, x)$ is bounded by a function elementary in h_α, uniformly in α.

- $\alpha = 0$
  This is trivial, since h_0 is elementary.
- $\alpha = \beta + 1$
  $T_h(\beta + 1, 1, x) = T_h(\beta, x, x)$, and by what proved above $T_h(\beta, x, x)$ is bounded by a function elementary in $h_{\beta+1} = h_\alpha$.
- α *a limit*
  $T_h(\alpha, 1, x) = T_h(\alpha_x, 1, x)$, and by induction hypothesis $T_h(\alpha_x, 1, x)$ is bounded by a function elementary in h_{α_x+1}, and hence in h_α (by VIII.9.14), uniformly in x. Then so is $T_h(\alpha, 1, x)$.

This proves that h_α is computable in time elementary in h_α, for any $\alpha < \omega(n)$ and any n, i.e. for any $\alpha < \epsilon_0$. □

Corollary VIII.9.19 *For any* $3 \leq \alpha \leq \epsilon_0$, $\mathcal{E}_\alpha$ *is a complexity class w.r.t. time and space.*

Proof. From VIII.9.18, as in VIII.7.7. □

The diagonal function ⋆

In this subsection we analyze the rate of growth of the diagonal function $d_{U(\epsilon_0)} = h_{\epsilon_0}$, defined in VIII.9.13, as we did for $d_{\mathcal{E}}$ in VIII.7.14, and for $d_{\mathcal{P}_1} = h_\omega$ in VIII.8.27.

Before turning to the computation of the position of h_{ϵ_0} in the Slow Growing Hierarchy defined in VIII.7.12, we briefly consider the Moderate Growing Hierarchy defined in VIII.8.22, and extend VIII.8.23 as follows.

Proposition VIII.9.20 Comparison of the Moderate and Fast Growing Hierarchies (Wainer [1972])

1. $h_\alpha = g_{\omega^\alpha}$, *for each* $\alpha \leq \epsilon_0$
2. $h_{\epsilon_0} = g_{\epsilon_0}$.

Proof. Part 1 is proved by induction on $\alpha \leq \epsilon_0$, as in VIII.8.23. In particular, if α is a limit, then

$$\begin{array}{rcll} g_{\omega^\alpha}(x) & = & g_{(\omega^\alpha)_x}(x) & \text{by definition of } g_{\omega^\alpha} \\ & = & g_{\omega^{\alpha_x}}(x) & \text{by definition of } (\omega^\alpha)_x \\ & = & h_{\alpha_x}(x) & \text{by induction hypothesis} \\ & = & h_\alpha(x) & \text{by definition of } h_\alpha. \end{array}$$

Part 2 follows from Part 1, since $g_{\omega^{\epsilon_0}} = h_{\epsilon_0}$, and ϵ_0 is a fixed-point of the ordinal exponential function, i.e. $\omega^{\epsilon_0} = \epsilon_0$. $\square$

Thus the Intermediate and Fast Growing Hierarchies meet at $\epsilon_0 = \phi_1(0)$ for the first time, and in general at every α such that $\alpha = \phi_1(\beta)$ for some β, i.e. at every α which is a fixed-point of the ordinal exponential function.

Exercises VIII.9.21 A refinement of the Extended Grzegorczyk Hierarchy. (Wainer [1972]) Define $\mathcal{G}_\alpha$ as in VIII.8.24.

a) $\mathcal{G}_\alpha \subset \mathcal{G}_\beta$, *if* $2 \leq \alpha < \beta \leq \epsilon_0$. (Hint: as in VIII.8.24.a, prove by induction on β that g_β strictly dominates g_α.)

b) $\mathcal{G}_{\omega^\alpha} = \mathcal{E}_\alpha$, *if* $3 \leq \alpha \leq \epsilon_0$. (Hint: as in VIII.8.24.b.)

c) $\mathcal{G}_{\epsilon_0} = \mathcal{E}_{\epsilon_0}$.

We now turn to the finer Slow Growing Hierarchy, with the goal of computing the position of h_{ϵ_0} in it. We might expect to go an immensely long way, since it takes $\epsilon_0 = \phi_1(0)$ steps to get to h_3, and $\phi_\omega(0)$ steps to get to h_ω.

The intuition that takes us to the ordinal we want is as follows. To obtain $f_{\phi_\omega(0)} \equiv_{\mathcal{E}} h_\omega$, we iterated the process of taking fixed-points ω times, starting from the ordinal exponential function. Since $h_{\alpha+1}$ is the iteration of h_α, to obtain $h_{\omega+1}$ we will have to iterate the process that took us to $\phi_\omega(0)$, i.e. to consider 'second-order' fixed-points, obtained by taking fixed-points of the process of taking usual fixed-points.

Technically, these 'second-order' fixed-points will be obtained in two steps. First, we extend the Veblen Hierarchy into the transfinite, thus defining ϕ_α for every countable ordinal α, and obtaining a doubly infinite matrix of countable ordinals, in which $\phi_\alpha(\gamma)$ is the γ-th element of the α-th row. Second, we turn to the consideration of the first column, and let

$$\phi_\Omega(\alpha) = \phi_\alpha(0).$$

Then we start the previous process again, letting $\phi_{\Omega+1}$ be the sequence of fixed-points of ϕ_Ω, and so on. In particular, $\phi_{\Omega+1}(0)$ is the smallest fixed-point of ϕ_Ω, i.e. the smallest α such that $\alpha = \phi_\Omega(\alpha) = \phi_\alpha(0)$; such an ordinal is called

Γ_0,[15] and (by the observation after VIII.8.25) it can be defined directly as

$$\begin{array}{rcl} \Gamma_0 & = & \lim_{x\in\omega} \phi_\Omega^{(x)}(0) & = & \lim\{0, \phi_0(0), \phi_{\phi_0(0)}(0), \phi_{\phi_{\phi_0(0)}(0)}(0), \ldots\} \\ & & & = & \lim\{0, 1, \epsilon_0, \phi_{\epsilon_0}(0), \ldots\}. \end{array}$$

The intuition above then amounts to saying that $h_{\omega+1} \equiv_{\mathcal{E}} f_{\Gamma_0}$.

We are thus led to the following definition, which extends VIII.8.25.

Definition VIII.9.22 The Veblen-Bachmann Hierarchy (Veblen [1908], Bachmann [1950]) *Let Ω be the first uncountable ordinal. The sequence $\{\phi_\alpha\}_\alpha$ of ordinal functions is defined by induction on α, as follows:*

$$\begin{array}{rcl} \phi_0(\gamma) & = & \omega^\gamma \\ \phi_{\alpha+1}(\gamma) & = & \textit{the } \gamma\textit{-th fixed-point of } \phi_\alpha \\ & = & \textit{the } \gamma\textit{-th ordinal } \beta \textit{ such that } \beta = \phi_\alpha(\beta) \\ \phi_\alpha(\gamma) & = & \begin{cases} \lim_{x\in\omega} \phi_{\alpha_x}(\gamma) & \textit{if } \alpha \textit{ is a limit and } cf(\alpha) = \omega \\ \phi_{\alpha_\gamma+1}(0) & \textit{if } \alpha \textit{ is a limit and } cf(\alpha) = \Omega, \end{cases} \end{array}$$

where, for α a limit, $cf(\alpha)$ is the **cofinality** *of α, i.e. the smallest ordinal θ such that α is the limit of an increasing sequence of ordinals of length θ,*[16] *and $\{\alpha_\gamma\}_{\gamma<cf(\alpha)}$ is any such sequence (called a* **fundamental sequence** *for α).*

We can spell out the definition just given, as follows:

- $\phi_{\alpha+1}(0) = \lim_{x\in\omega} \phi_\alpha^{(x)}(0)$
- $\phi_{\alpha+1}(\gamma+1) = \lim_{x\in\omega} \phi_\alpha^{(x)}(\phi_{\alpha+1}(\gamma)+1)$
- $\phi_{\alpha+1}(\gamma) = \lim_{x\in\omega} \phi_{\alpha+1}(\gamma_x)$, *if γ is a limit*
- $\phi_\alpha(\gamma) = \lim_{x\in\omega} \phi_{\alpha_x}(\gamma)$, *if α is a limit and $cf(\alpha) = \omega$*
- $\phi_\alpha(\gamma) = \lim_{x\in\omega} \phi_{\alpha_\gamma}^{(x)}(0)$, *if α is a limit and $cf(\alpha) = \Omega$.*

The first three expressions have already been noted after VIII.8.25 (the proof has been given there for $\alpha < \omega$, but it holds in general), while the fourth one holds by definition, and extends the definition of $\phi_\omega(0)$ given in VIII.8.25.

[15] Γ_0 is quite important in Proof Theory. For example, Feferman [1964] has proved that it is the smallest ordinal that is not predicatively definable.

[16] The definition of cofinality can be extended to all ordinals, by letting $cf(0) = 0$ and $cf(\alpha+1) = 1$. Notice that the cofinality of an ordinal is always a cardinal, and thus the smallest infinite cofinalities are ω (e.g. for every countable limit ordinal) and Ω.

The last expression follows from the first one, since for a limit ordinal α of cofinality Ω, we have defined $\phi_\alpha(\gamma) = \phi_{\alpha_\gamma+1}(0)$. This is actually the reason for the definition, which departs slightly from the example of ϕ_Ω given above, according to which we should have let $\phi_\alpha(\gamma) = \phi_{\alpha_\gamma}(0)$. However, by doing this it would be difficult to express $\phi_\alpha(\gamma)$ as the limit of a sequence (which we want to do, to be able to define fundamental sequences), because we would not know whether α_γ is a successor or a limit ordinal, and of which cofinality in the latter case. With the new definition, we always work with successor ordinals, which we know how to handle.

Recall that every ordinal $\alpha \neq 0$ has a unique expansion to the base Ω, of the form

$$\alpha = \Omega^{\beta_1} \cdot \gamma_1 + \cdots + \Omega^{\beta_m} \cdot \gamma_m,$$

where $\gamma_1, \ldots, \gamma_m$ are countable ordinals and $\beta_m < \cdots < \beta_1$ (the **Cantor normal form** of α to the base Ω). The Cantor normal form of α is called *pure* if (inductively) all γ_i are finite, and all β_i are finite or in pure Cantor normal form. Thus a pure Cantor normal form is a term built up from Ω and finite ordinals by ordinal sum, product and exponentiation.

The notation ϵ_0 for (the first fixed-point of ϕ_0, i.e.) $\phi_1(0)$ is extended by letting $\epsilon_\alpha = \phi_1(\alpha)$. Thus $\epsilon_\Omega = \Omega$, since $\omega^\Omega = \Omega$, and there are uncountably many countable ordinals α such that $\omega^\alpha = \alpha$. Moreover, $\epsilon_{\Omega+1}$ is the first ϵ-number greater than Ω, i.e. the smallest α such that $\alpha = \Omega^\alpha$.

If $\alpha < \epsilon_{\Omega+1}$, then in the Cantor normal form of α to the base Ω we have $\beta_1 < \alpha$, and we can use the expansion in inductive definitions. From the representation just recalled, it follows that every limit ordinal $\alpha < \epsilon_{\Omega+1}$ can be uniquely written in one of the forms

$$\beta + \Omega^0 \cdot \delta,\ \delta \text{ a limit} \qquad \text{or} \qquad \beta + \Omega^{\gamma+1} \cdot \delta \qquad \text{or} \qquad \beta + \Omega^\gamma \cdot \delta,\ \gamma \text{ a limit}$$

with δ countable and $\gamma < \alpha$. This justifies the next definition, which is the analogue of VIII.7.13 one level up.

Definition VIII.9.23 Fundamental Sequences up to $\epsilon_{\Omega+1}$ (Howard [1972]) *If $\alpha < \epsilon_{\Omega+1}$ is a limit ordinal, then:*

1. *if $cf(\alpha) = \omega$ and $x < \omega$:*

$$\alpha_x = \begin{cases} \Omega^\gamma \cdot \delta_x & \text{if } \alpha = \Omega^\gamma \cdot \delta \text{ and } \delta \text{ is a limit} \\ \Omega^{\gamma_x} & \text{if } \alpha = \Omega^\gamma,\ \gamma \text{ is a limit and } cf(\gamma) = \omega \\ \beta + \gamma_x & \text{if } \alpha = \beta + \gamma,\ \gamma \text{ is a limit and } cf(\gamma) = \omega \end{cases}$$

2. *if $cf(\alpha) = \Omega$ and $\sigma < \Omega$:*

$$\alpha_\sigma = \begin{cases} \Omega^\gamma \cdot \sigma & \text{if } \alpha = \Omega^{\gamma+1} \\ \Omega^{\gamma_\sigma} & \text{if } \alpha = \Omega^\gamma,\ \gamma \text{ is a limit and } cf(\gamma) = \Omega \\ \beta + \gamma_\sigma & \text{if } \alpha = \beta + \gamma,\ \gamma \text{ is a limit and } cf(\gamma) = \Omega. \end{cases}$$

Moreover,

$$(\epsilon_{\Omega+1})_0 = \Omega \qquad \textit{and} \qquad (\epsilon_{\Omega+1})_{x+1} = \Omega^{(\epsilon_{\Omega+1})_x}.$$

The previous definition does introduce fundamental sequences for *some* uncountable limit ordinals (e.g. for Ω itself, in which case $\Omega_\sigma = \sigma$), but certainly not for *every* limit ordinal less than $\epsilon_{\Omega+1}$, since we do not have fundamental sequences for every *countable* limit ordinal.

However, notice that every ordinal $\alpha \neq 0$ has a unique expansion of the form

$$\alpha = \phi_{\alpha_1}(\beta_1) \cdot x_1 + \cdots + \phi_{\alpha_m}(\beta_m) \cdot x_m,$$

where $x_1, \ldots, x_m$ are integers, $\phi_{\alpha_m}(\beta_m) < \cdots < \phi_{\alpha_1}(\beta_1)$ and $\alpha_m \leq \cdots \leq \alpha_1$.

Notice also that if $\alpha < \phi_{\epsilon_{\Omega+1}+1}(0)$, then in the above expansion of α we have either $\alpha_1 < \epsilon_{\Omega+1}$, or $\alpha_1 = \epsilon_{\Omega+1}$ and $\beta_1 < \alpha_1$, and we can use the expansion in inductive definitions. This justifies the following extension of VIII.7.13 and VIII.8.26.

Definition VIII.9.24 Fundamental Sequences up to $\phi_{\epsilon_{\Omega+1}+1}(0)$. *If $\alpha < \phi_{\epsilon_{\Omega+1}+1}(0)$ is a limit ordinal, then:*

$$\alpha_x = \begin{cases} \omega^\gamma \cdot x & \textit{if } \alpha = \phi_0(\gamma+1) \\ \omega^{\gamma_x} & \textit{if } \alpha = \phi_0(\gamma) \textit{ and } \gamma \textit{ is a limit} \\ \phi_\beta^{(x)}(0) & \textit{if } \alpha = \phi_{\beta+1}(0) \\ \phi_\beta^{(x)}(\phi_{\beta+1}(\gamma)+1) & \textit{if } \alpha = \phi_{\beta+1}(\gamma+1) \\ \phi_{\beta+1}(\gamma_x) & \textit{if } \alpha = \phi_{\beta+1}(\gamma) \textit{ and } \gamma \textit{ is a limit} \\ \phi_{\beta_x}(\gamma) & \textit{if } \alpha = \phi_\beta(\gamma),\ \beta \textit{ is a limit and } cf(\beta) = \omega \\ \phi_{\beta_\gamma}^{(x)}(0) & \textit{if } \alpha = \phi_\beta(\gamma),\ \beta \textit{ is a limit and } cf(\beta) = \Omega \\ \beta + \gamma_x & \textit{if } \alpha = \beta + \gamma \textit{ and } \gamma \textit{ is a limit.} \end{cases}$$

Moreover,

$$(\phi_{\epsilon_{\Omega+1}+1}(0))_x = \phi_{\epsilon_{\Omega+1}}^{(x)}(0).$$

In the following, we use the fundamental sequences just defined in the definition VIII.7.12 of f_α, when $\alpha \leq \phi_{\epsilon_{\Omega+1}+1}(0)$.

Since $\phi_\Omega(\omega) = \phi_{\omega+1}(0)$ by definition, VIII.8.27 proved that $h_\omega \equiv_{\mathcal{E}} f_{\phi_\Omega(\omega)}$. This is generalized in Part 1 of the next result.

Since $\Gamma_0 = \phi_{\Omega+1}(0)$, the intuition discussed before VIII.9.22 amounts to saying that $h_{\omega+1} \equiv_{\mathcal{E}} f_{\phi_{\Omega+1}(0)}$. This is proved and generalized in Part 2 of the next result.

Theorem VIII.9.25 Comparison of the Slow and Fast Growing Hierarchies (Girard [1981]) *Let $L(\alpha)$ (where 'L' stands for 'lifting') be the ordinal obtained by first writing α in pure Cantor normal form to the base ω, and then substituting Ω for ω everywhere. Then:*

1. *$h_\alpha \equiv_{\mathcal{E}} f_{\phi_{L(\alpha)}(\omega)}$, for $\omega \leq \alpha < \epsilon_0$ and α a limit*
2. *$h_\alpha \equiv_{\mathcal{E}} f_{\phi_{L(\alpha)}(0)}$, for $\omega < \alpha < \epsilon_0$ and α a successor*
3. *$h_{\epsilon_0} \equiv_{\mathcal{E}} f_{\phi_{\epsilon_{\Omega+1}+1}(0)}$.*

Proof. We extend the correspondence among finite levels of the Veblen and the Fast Growing Hierarchies introduced in the proof of VIII.8.27 into the transfinite, as follows:

ordinals	numbers
ordinal exponentiation	numerical exponentiation
fixed-points	iteration
second-order fixed-points	diagonalization

As in VIII.8.27, we mimic the definition of fundamental sequences required in the definition of $\{h_\alpha\}_{\alpha \leq \epsilon_0}$ and let, for $\alpha \leq \epsilon_0$:

$$\begin{aligned} H_0(x,y) &= x^y \\ H_{\alpha+1}(x,0) &= H_\alpha^{(x)}(x,0) \\ H_{\alpha+1}(x,y+1) &= H_\alpha^{(x)}(x, H_{\alpha+1}(x,y)+1) \\ H_\alpha(x,y) &= H_{\alpha_y+1}(x,0) \text{ if } \alpha \text{ is a limit.} \end{aligned}$$

Recall from VIII.8.27 that the iteration of H_α is performed on its second variable:

$$\begin{aligned} H_\alpha^{(0)}(x,y) &= y \\ H_\alpha^{(z+1)}(x,y) &= H_\alpha(x, H_\alpha^{(z)}(x,y)). \end{aligned}$$

It is quite clear that the following are elementarily equivalent:

- $H_\alpha(x,0)$ and $h_{2+\alpha}(x)$, if α is a successor
- $H_\alpha(x,x)$ and $h_{2+\alpha}(x)$, if α is a limit.

This follows by induction: $H_0(x,0) = 1$ is elementary (as is h_2); $H_{\alpha+1}$ is obtained from H_α by iteration (notice that H_0 is of exponential growth, as is h_2); and H_α for α a limit is obtained by diagonalization of $\{H_{\alpha_x+1}\}_{x\in\omega}$, while h_α is obtained by diagonalization of $\{h_{\alpha_x}\}_{x\in\omega}$ (which is elementarily equivalent to the diagonalization of $\{h_{\alpha_x+1}\}_{x\in\omega}$).

Similarly, we mimic the definition of fundamental sequences required in the definition of $\{f_\alpha\}_{\alpha \leq \phi_{\epsilon_{\Omega+1}+1}}$ and let, for $\alpha \leq \epsilon_{\Omega+1}$:

$$F_0(x,y) = x^y$$

$$\begin{aligned}
F_{\alpha+1}(x,0) &= F_\alpha^{(x)}(x,0)\\
F_{\alpha+1}(x,y+1) &= F_\alpha^{(x)}(x,F_{\alpha+1}(x,y)+1)\\
F_\alpha(x,y) &= F_{\alpha_x}(x,y) \text{ if } \alpha \text{ is a limit and } cf(\alpha)=\omega\\
F_\alpha(x,y) &= F_{\alpha_y+1}(x,0) \text{ if } \alpha \text{ is a limit and } cf(\alpha)=\Omega.
\end{aligned}$$

The connection between $\{H_\alpha\}_{\alpha<\epsilon_0}$ and $\{F_\alpha\}_{\alpha<\epsilon_{\Omega+1}}$ is that, for α in pure Cantor normal form to the base ω,

$$H_\alpha(x,y) = F_{L(\alpha)}(x,y). \qquad \text{(VIII.12)}$$

Indeed, the definitions of H_α and $F_{L(\alpha)}$ are parallel for successor ordinals, and for limit ordinals of respective cofinalities ω and Ω. L changes ordinals of cofinality ω into ordinals of cofinality Ω, by preserving pure Cantor normal forms (in particular, successors and fundamental sequences).

Extending the work done in VIII.8.27 we show that, for $\phi_\alpha(\beta) < \phi_{\epsilon_{\Omega+1}+1}(0)$,

$$f_{\phi_\alpha(\beta)}(x) = F_\alpha(x,f_\beta(x)). \qquad \text{(VIII.13)}$$

Then Parts 1 and 2 follow:

1. $\alpha < \epsilon_0$ *a limit*

$$\begin{aligned}
H_\alpha(x,x) &= H_\alpha(x,f_\omega(x)) && \text{by definition of } f_\omega\\
&= F_{L(\alpha)}(x,f_\omega(x)) && \text{by VIII.12}\\
&= f_{\phi_{L(\alpha)}(\omega)}(x) && \text{by VIII.13,}
\end{aligned}$$

and thus

$$h_{2+\alpha} \equiv_{\mathcal{E}} f_{\phi_{L(\alpha)}(\omega)}.$$

2. $\alpha < \epsilon_0$ *a successor*

$$\begin{aligned}
H_\alpha(x,0) &= H_\alpha(x,f_0(x)) && \text{by definition of } f_0\\
&= F_{L(\alpha)}(x,f_0(x)) && \text{by VIII.12}\\
&= f_{\phi_{L(\alpha)}(0)}(x) && \text{by VIII.13,}
\end{aligned}$$

and thus

$$h_{2+\alpha} \equiv_{\mathcal{E}} f_{\phi_{L(\alpha)}(0)}.$$

Parts 1 and 2 now follow because $2+\alpha = \alpha$ for $\alpha \geq \omega$.

We prove VIII.13 by induction on α. The cases $\alpha = 0$ and α a successor are as in VIII.8.27 (using the first three defining equations for F_α). For α a limit we have two cases:

- $cf(\alpha) = \omega$

$$\begin{array}{rcll} f_{\phi_\alpha(\beta)}(x) & = & f_{(\phi_\alpha(\beta))_x}(x) & \text{by definition of } f_{\phi_\alpha(\beta)} \\ & = & f_{\phi_{\alpha_x}(\beta)}(x) & \text{by definition of } (\phi_\alpha(\beta))_x \\ & = & F_{\alpha_x}(x, f_\beta(x)) & \text{by induction hypothesis} \\ & = & F_\alpha(x, f_\beta(x)) & \text{by definition of } F_\alpha. \end{array}$$

- $cf(\alpha) = \Omega$

$$\begin{array}{rcll} f_{\phi_\alpha(\beta)}(x) & = & f_{(\phi_\alpha(\beta))_x}(x) & \text{by definition of } f_{\phi_\alpha(\beta)} \\ & = & f_{\phi^{(x)}_{\alpha_\beta}(0)}(x) & \text{by definition of } (\phi_\alpha(\beta))_x \\ & = & F^{(x)}_{\alpha_\beta}(x, f_0(x)) & \text{by induction hypothesis } (x \text{ times}) \\ & = & F^{(x)}_{\alpha_\beta}(x, 0) & \text{by definition of } f_0. \end{array}$$

If we show that, for α a limit and $cf(\alpha) = \Omega$,

$$(\forall x)(\forall y)[F_{\alpha_\beta}(x, y) \;=\; F_{\alpha_{f_\beta(x)}}(x, y)], \tag{VIII.14}$$

then we can continue the sequence of equalities, and obtain

$$\begin{array}{rcll} f_{\phi_\alpha(\beta)}(x) & = & F^{(x)}_{\alpha_{f_\beta(x)}}(x, 0) & \text{by VIII.14 } (x \text{ times}) \\ & = & F_{\alpha_{f_\beta(x)}+1}(x, 0) & \text{by definition of } F_{\alpha_{f_\beta(x)}+1} \\ & = & F_\alpha(x, f_\beta(x)) & \text{by definition of } F_\alpha. \end{array}$$

Showing VIII.14 requires some work, and will be done by using the following function $C(\gamma, x)$ (called the *collapse of* γ *at* x), defined by induction on γ:

$$\begin{array}{rcl} C(0, x) & = & 0 \\ C(\delta + \Omega^\beta \cdot \gamma, x) & = & C(\delta, x) + \Omega^{C(\beta, x)} \cdot f_\gamma(x). \end{array}$$

The collapse allows us to break the proof of VIII.14 into two steps. First, we show that the collapse of γ preserves F_γ, i.e. that

$$F_{C(\gamma,x)}(x, y) \;=\; F_\gamma(x, y). \tag{VIII.15}$$

Second, we prove that, for α a limit and $cf(\alpha) = \Omega$,

$$C(\alpha_\beta, x) \;=\; C(\alpha_{f_\beta(x)}, x). \tag{VIII.16}$$

VIII.14 follows from VIII.15 and VIII.16, for α a limit and $cf(\alpha) = \Omega$:

$$\begin{array}{rcll} F_{\alpha_\beta}(x, y) & = & F_{C(\alpha_\beta, x)}(x, y) & \text{by VIII.15} \\ & = & F_{C(\alpha_{f_\beta(x)}, x)}(x, y) & \text{by VIII.16} \\ & = & F_{\alpha_{f_\beta(x)}}(x, y) & \text{by VIII.15.} \end{array}$$

It remains to prove VIII.15 and VIII.16.

We first deal with VIII.15 and prove that

$$F_{C(\gamma,x)}(x,y) = F_\gamma(x,y)$$

by induction on γ:

- $\gamma = 0$
  This is trivial, since $C(0,x) = 0$.

- $\gamma + 1$
  It is enough to show that

$$C(\gamma+1,x) \quad = \quad C(\gamma,x)+1. \tag{VIII.17}$$

  Then we can proceed by induction on y, as follows:

$$\begin{array}{lll} & F_{C(\gamma+1,x)}(x,0) & \\ = & F_{C(\gamma,x)+1}(x,0) & \text{by VIII.17} \\ = & F^{(x)}_{C(\gamma,x)}(x,0) & \text{by definition of } F_{C(\gamma,x)+1} \\ = & F^{(x)}_{\gamma}(x,0) & \text{by induction hypothesis on } \gamma \\ = & F_{\gamma+1}(x,0) & \text{by definition of } F_{\gamma+1} \end{array}$$

  and

$$\begin{array}{lll} & F_{C(\gamma+1,x)}(x,y+1) & \\ = & F_{C(\gamma,x)+1}(x,y+1) & \text{by VIII.17} \\ = & F^{(x)}_{C(\gamma,x)}(x,F_{C(\gamma,x)+1}(x,y)+1) & \text{by definition of } F_{C(\gamma,x)+1} \\ = & F^{(x)}_{C(\gamma,x)}(x,F_{C(\gamma+1,x)}(x,y)+1) & \text{by VIII.17} \\ = & F^{(x)}_{C(\gamma,x)}(x,F_{\gamma+1}(x,y)+1) & \text{by induction hypothesis on } y \\ = & F^{(x)}_{\gamma}(x,F_{\gamma+1}(x,y)+1) & \text{by induction hypothesis on } \gamma \\ = & F_{\gamma+1}(x,y+1) & \text{by definition of } F_{\gamma+1}. \end{array}$$

- γ a *limit*
  If $cf(\gamma) = \omega$, it is enough to show that

$$C(\gamma,x) \quad = \quad C(\gamma_x,x), \tag{VIII.18}$$

  since then

$$\begin{array}{rcll} F_{C(\gamma,x)}(x,y) & = & F_{C(\gamma_x,x)}(x,y) & \text{by VIII.18} \\ & = & F_{\gamma_x}(x,y) & \text{by induction hypothesis} \\ & = & F_\gamma(x,y) & \text{by definition of } F_\gamma. \end{array}$$

If $cf(\gamma) = \Omega$, then $cf(C(\gamma, x)) = \Omega$,[17] and it is enough to show that

$$(C(\gamma, x))_y = C(\gamma_y, x), \tag{VIII.19}$$

since then

$$\begin{array}{rcll} F_{C(\gamma,x)}(x, y) & = & F_{(C(\gamma,x))_y+1}(x, 0) & \text{by definition of } F_{C(\gamma,x)} \\ & = & F_{C(\gamma_y,x)+1}(x, 0) & \text{by VIII.19} \\ & = & F_{C(\gamma_y+1,x)}(x, 0) & \text{by VIII.17} \\ & = & F_{\gamma_y+1}(x, 0) & \text{by induction hypothesis} \\ & = & F_\gamma(x, y) & \text{by definition of } F_\gamma. \end{array}$$

We now turn to the three properties VIII.17, VIII.18 and VIII.19, which will allow us to conclude the proof of VIII.15. In the proofs, we consider the Cantor normal form of γ to the base Ω and concentrate on the smallest exponent of Ω, i.e. we let

$$\gamma = \sigma + \Omega^\beta \cdot \delta,$$

with δ countable and σ in Cantor normal form to the base Ω, with all exponents greater than β.

- *$C(\gamma + 1, x) = C(\gamma, x) + 1$* (VIII.17)
  If $\beta = 0$, then $\gamma + 1 = \sigma + \Omega^0(\delta + 1)$, and so

$$\begin{array}{rll} & C(\gamma + 1, x) & \\ = & C(\sigma, x) + \Omega^{C(0,x)} \cdot f_{\delta+1}(x) & \text{by definition of } C \\ = & C(\sigma, x) + \Omega^{C(0,x)} \cdot (f_\delta(x) + 1) & \text{by definition of } f_{\delta+1} \\ = & C(\sigma, x) + \Omega^{C(0,x)} \cdot f_\delta(x) + \Omega^{C(0,x)} & \\ = & C(\gamma, x) + \Omega^{C(0,x)} & \text{by definition of } C \\ = & C(\gamma, x) + 1 & \text{because } C(0, x) = 0. \end{array}$$

  If $\beta > 0$, then $\gamma + 1 = \gamma + \Omega^0 \cdot 1$, and so

$$\begin{array}{rll} & C(\gamma + 1, x) & \\ = & C(\gamma, x) + \Omega^{C(0,x)} \cdot f_1(0) & \text{by definition of } C \\ = & C(\gamma, x) + f_1(0) & \text{because } C(0, x) = 0 \\ = & C(\gamma, x) + 1 & \text{by definition of } f_1. \end{array}$$

- *$C(\gamma, x) = C(\gamma_x, x)$, if γ is a limit and $cf(\gamma) = \omega$* (VIII.18)
  If δ is a limit, then

$$\begin{array}{rll} & C(\gamma, x) & \\ = & C(\sigma, x) + \Omega^{C(\beta,x)} \cdot f_\delta(x) & \text{by definition of } C \\ = & C(\sigma, x) + \Omega^{C(\beta,x)} \cdot f_{\delta_x}(x) & \text{by definition of } f_\delta \\ = & C(\sigma + \Omega^\beta \cdot \delta_x, x) & \text{by definition of } C \\ = & C(\gamma_x, x) & \text{by definition of } \gamma_x. \end{array}$$

[17] See the comments after the end of the proof.

If δ is a successor, β is a limit and $cf(\beta)=\omega$, then $\gamma=\sigma+\Omega^\beta\cdot(\delta-1)+\Omega^\beta$, and

$$\begin{array}{lll} & C(\gamma,x) & \\ = & C(\sigma,x)+\Omega^{C(\beta,x)}\cdot f_\delta(x) & \text{by definition of } C \\ = & C(\sigma,x)+\Omega^{C(\beta,x)}\cdot(f_{\delta-1}(x)+1) & \text{by definition of } f_\delta \\ = & C(\sigma,x)+\Omega^{C(\beta,x)}\cdot f_{\delta-1}(x)+\Omega^{C(\beta,x)} & \\ = & C(\sigma,x)+\Omega^{C(\beta,x)}\cdot f_{\delta-1}(x)+\Omega^{C(\beta_x,x)} & \text{by induction hypothesis} \\ = & C(\sigma+\Omega^\beta\cdot(\delta-1)+\Omega^{\beta_x},x) & \text{by definition of } C \\ = & C(\gamma_x,x) & \text{by definition of } \gamma_x. \end{array}$$

- $(C(\gamma,x))_y=C(\gamma_y,x)$, *if* γ *is a limit and* $cf(\gamma)=\Omega$ (VIII.19)
  If δ is a successor, β is a limit and $cf(\beta)=\Omega$, then $\gamma=\sigma+\Omega^\beta\cdot(\delta-1)+\Omega^\beta$, $cf(C(\beta,x))=\Omega$,[18] and

$$\begin{array}{lll} & (C(\gamma,x))_y & \\ = & (C(\sigma+\Omega^\beta\cdot(\delta-1),x)+\Omega^{C(\beta,x)})_y & \text{by definition of } C \\ = & C(\sigma+\Omega^\beta\cdot(\delta-1),x)+(\Omega^{C(\beta,x)})_y & \text{by definition of } (\ )_y \\ = & C(\sigma+\Omega^\beta\cdot(\delta-1),x)+\Omega^{(C(\beta,x))_y} & \text{by definition of } (\ )_y \\ = & C(\sigma+\Omega^\beta\cdot(\delta-1),x)+\Omega^{C(\beta_y,x)} & \text{by induction hypothesis} \\ = & C(\sigma+\Omega^\beta(\delta-1)+\Omega^{\beta_y},x) & \text{by definition of } C \\ = & C(\gamma_y,x) & \text{by definition of } \gamma_y. \end{array}$$

If δ and β are successors, then $\gamma=\sigma+\Omega^\beta\cdot(\delta-1)+\Omega^\beta$, and

$$\begin{array}{lll} & (C(\gamma,x))_y & \\ = & (C(\sigma+\Omega^\beta\cdot(\delta-1),x)+\Omega^{C(\beta,x)})_y & \text{by definition of } C \\ = & C(\sigma+\Omega^\beta\cdot(\delta-1),x)+(\Omega^{C(\beta,x)})_y & \text{by definition of } (\ )_y \\ = & C(\sigma+\Omega^\beta\cdot(\delta-1),x)+(\Omega^{(C(\beta-1,x)+1)})_y & \text{by VIII.17} \\ = & C(\sigma+\Omega^\beta\cdot(\delta-1),x)+\Omega^{C(\beta-1,x)}\cdot y & \text{by definition of } (\ )_y \\ = & C(\sigma+\Omega^\beta\cdot(\delta-1),x)+\Omega^{C(\beta-1,x)}\cdot f_y(x) & \text{by definition of } f_y \\ = & C(\sigma+\Omega^\beta\cdot(\delta-1)+\Omega^{\beta-1}\cdot y,x) & \text{by definition of } C \\ = & C(\gamma_y,x) & \text{by definition of } \gamma_y. \end{array}$$

We have thus proved VIII.17, VIII.18 and VIII.19, and this concludes the proof of VIII.15.

We now deal with VIII.16, and prove that

$$C(\alpha_\beta,x)=C(\alpha_{f_\beta(x)},x)$$

for α a limit and $cf(\alpha)=\Omega$. We have two cases:

[18]See the comments after the end of the proof.

- $\alpha = \sigma + \Omega^{\gamma+1}$

$$\begin{array}{rcll} C(\alpha_\beta, x) & = & C(\sigma + \Omega^\gamma \cdot \beta, x) & \text{by definition of } \alpha_\beta \\ & = & C(\sigma, x) + \Omega^{C(\gamma,x)} \cdot f_\beta(x) & \text{by definition of } C \\ & = & C(\sigma, x) + \Omega^{C(\gamma,x)} \cdot f_{f_\beta(x)}(x) & \text{because } f_n(x) = n \\ & = & C(\sigma + \Omega^\gamma \cdot f_\beta(x), x) & \text{by definition of } C \\ & = & C(\alpha_{f_\beta(x)}, x) & \text{by definition of } \alpha_{f_\beta(x)}. \end{array}$$

- $\alpha = \sigma + \Omega^\gamma$, γ *a limit and* $cf(\gamma) = \Omega$

$$\begin{array}{rcll} C(\alpha_\beta, x) & = & C(\sigma + \Omega^{\gamma_\beta}, x) & \text{by definition of } \alpha_\beta \\ & = & C(\sigma, x) + \Omega^{C(\gamma_\beta,x)} & \text{by definition of } C \\ & = & C(\sigma, x) + \Omega^{C(\gamma_{f_\beta(x)},x)} & \text{by induction hypothesis} \\ & = & C(\sigma + \Omega^{\gamma_{f_\beta(x)}}, x) & \text{by definition of } C \\ & = & C(\alpha_{f_\beta(x)}, x) & \text{by definition of } \alpha_{f_\beta(x)}. \end{array}$$

It only remains to prove Part 3. Clearly

$$H_{\epsilon_0}(x, x) = H_{(\epsilon_0)_x+1}(x, 0)$$

is elementarily equivalent to h_{ϵ_0}, and

$$\begin{array}{rcll} H_{\epsilon_0}(x, x) & = & H_{(\epsilon_0)_x+1}(x, 0) & \text{by definition of } H_{\epsilon_0} \\ & = & F_{L((\epsilon_0)_x+1)}(x, 0) & \text{by VIII.12} \\ & = & F_{(\epsilon_{\Omega+1})_x+1}(x, 0) & \text{by definition of } L \\ & = & F^{(x)}_{(\epsilon_{\Omega+1})_x}(x, 0) & \text{by definition of } F_{(\epsilon_{\Omega+1})_x+1} \end{array}$$

If we show that, for each $n > 0$,

$$F^{(n)}_{(\epsilon_{\Omega+1})_x}(x, 0) \quad = \quad f_{\phi^{(n)}_{\epsilon_{\Omega+1}}(0)}(x), \tag{VIII.20}$$

then we can continue the sequence of equalities, and obtain

$$\begin{array}{rcll} H_{\epsilon_0}(x, x) & = & f_{\phi^{(x)}_{\epsilon_{\Omega+1}}(0)}(x) & \text{by VIII.20} \\ & = & f_{(\phi_{\epsilon_{\Omega+1}+1}(0))_x}(x) & \text{by definition of } (\ )_x \\ & = & f_{\phi_{\epsilon_{\Omega+1}+1}(0)}(x) & \text{by definition of } f_{\phi_{\epsilon_{\Omega+1}+1}(0)}. \end{array}$$

We can prove VIII.20 by induction on $n > 0$, as follows:

$$\begin{array}{rcll} F_{(\epsilon_{\Omega+1})_x}(x, 0) & = & F_{(\epsilon_{\Omega+1})_x}(x, f_0(x)) & \text{by definition of } f_0 \\ & = & f_{\phi_{(\epsilon_{\Omega+1})_x}(0)}(x) & \text{by VIII.13} \\ & = & f_{(\phi_{\epsilon_{\Omega+1}}(0))_x}(x) & \text{by definition of } (\ )_x \\ & = & f_{\phi_{\epsilon_{\Omega+1}}(0)}(x) & \text{by definition of } f_{\phi_{\epsilon_{\Omega+1}}(0)} \end{array}$$

and

$$\begin{array}{rcll}
F^{(n+1)}_{(\epsilon_{\Omega+1})_x}(x,0) & = & F_{(\epsilon_{\Omega+1})_x}(x, F^{(n)}_{(\epsilon_{\Omega+1})_x}(x,0)) & \\
& = & F_{(\epsilon_{\Omega+1})_x}(x, f_{\phi^{(n)}_{\epsilon_{\Omega+1}}(0)}(x)) & \text{by induction hypothesis} \\
& = & f_{\phi_{(\epsilon_{\Omega+1})_x}(\phi^{(n)}_{\epsilon_{\Omega+1}}(0))}(x) & \text{by VIII.13} \\
& = & f_{(\phi_{\epsilon_{\Omega+1}}(\phi^{(n)}_{\epsilon_{\Omega+1}}(0)))_x}(x) & \text{by definition of } (\ )_x \\
& = & f_{(\phi^{(n+1)}_{\epsilon_{\Omega+1}}(0))_x}(x) & \\
& = & f_{\phi^{(n+1)}_{\epsilon_{\Omega+1}}(0)}(x) & \text{by definition of } f_{\phi^{(n+1)}_{\epsilon_{\Omega+1}}(0)}.
\end{array}$$

We have thus proved VIII.20, and this concludes the proof of Part 3. □

This proof hides a small technical point, which is better made explicit. In a few places, we have used a fact that looks quite innocent, namely

$$cf(\gamma) = \Omega \Rightarrow cf(C(\gamma, x)) = \Omega.$$

This would seem to be easy to verify, by induction on γ. Indeed, let $\gamma = \sigma + \Omega^\beta \cdot \delta$ where $cf(\gamma) = \Omega$, and consider the two possible cases:

- *δ and β successors*
  Then

$$\begin{array}{rcl}
C(\gamma, x) & = & C(\sigma, x) + \Omega^{C(\beta,x)} \cdot f_\delta(x) \\
& = & C(\sigma, x) + \Omega^{C(\beta-1,x)+1} \cdot f_\delta(x) \\
& = & C(\sigma, x) + \Omega^{C(\beta-1,x)} \cdot f_\delta(x) + \Omega \cdot f_\delta(x)
\end{array}$$

  by definition of C, and VIII.17.

- *δ a successor, β a limit and $cf(\beta) = \Omega$*
  Then

$$C(\gamma, x) = C(\sigma, x) + \Omega^{C(\beta,x)} \cdot f_\delta(x)$$

  by definition of C, and $cf(C(\beta, x)) = \Omega$ by induction hypothesis.

The problem is that the last case gives $cf(C(\gamma, x)) = \Omega$ only if we know that $C(\gamma, x)$ is in Cantor normal form, i.e. that $C(\beta, x)$ is the smallest exponent of Ω. But the definition of collapse uses the functions f_α's, and it might happen that $\alpha_0 < \alpha_1$ but $f_{\alpha_0}(x) > f_{\alpha_1}(x)$ (since f_{α_1} dominates f_{α_0} only from a certain point on). This might produce a reversal in the order of exponents of Ω, and $C(\gamma, x)$ might not be in Cantor normal form even if γ is.

Thus our proof of VIII.13 is not valid in general, but it does work under the additional hypothesis that the relevant collapses are in pure Cantor normal form. This is all we need for the result, since $L(\alpha)$ is in pure Cantor normal form by definition (having been obtained by substituting Ω for ω in a pure Cantor normal form to the base ω), and in this case the collapse does not change the ordinal (since all exponents are integers, and $f_n(x) = n$).

Other results and expositions about the comparison of the Slow and Fast Growing Hierarchies are in Schmerl [1982], Cichon and Wainer [1983], Dennis Jones and Wainer [1984], Wainer [1985], Weiermann [1995].

A natural stopping point $\star$

The theory of ϵ_0-recursive functions can be extended to any ordinal γ for which a natural and uniform system of notations exists. In particular, it is then possible to define the notion of γ-recursion, as well as to extend the definition of $\{h_\alpha\}_\alpha$ and $\{\mathcal{E}_\alpha\}_\alpha$ to all ordinals $\alpha < \gamma$. Appropriate conditions on the system of notations allow for a proof of the usual connections among such notions, such as VIII.9.15 and VIII.9.17. See Löb and Wainer [1970], [1970a], Schmidt [1976], [1977], Zemke [1977], Rose [1984], Buchholz, Cichon and Weiermann [1994], Möllerfeld and Weiermann [199?] for details.

Specific ordinals for which such natural and uniform systems of notations exist have been exhibited in Proof Theory, see Takeuti [1975], Schütte [1977], Buchholz, Feferman, Pohlers and Sieg [1981], Girard [1987], Buchholz and Schütte [1988], and Pohlers [1992]. Significant examples of such ordinals arise from systems of Second-Order Arithmetic obtained by extending Peano Arithmetic by various forms of comprehension axioms (asserting the existence of all sets defined by formulas of a given complexity):

- Δ^0_ω-comprehension: $\phi_1(\epsilon_0) = \epsilon_{\epsilon_0}$
- Δ^1_1-comprehension: $\phi_{\epsilon_0}(0)$
- Π^1_1-comprehension: $\phi_{\Omega_\omega \cdot \epsilon_0}(0)$
- Δ^1_2-comprehension: $\phi_{\Omega_{\epsilon_0}}(0)$.

Other significant examples arise from systems based on different principles:

- Predicative Analysis: $\phi_{\Omega+1}(0) = \Gamma_0$
- Kripke-Plated Set Theory with Infinity: $\phi_{\epsilon_{\Omega+1}}(0)$
- α-Iterated Inductive Definitions: $\phi_{\epsilon_{\Omega_\alpha+1}}(0)$.

The definitions of the ordinals just quoted require appropriate extensions of the Veblen-Bachmann Hierarchy (VIII.8.25 and VIII.9.22).

Extensions of VIII.9.25 have been considered by Girard [1981]. Notice that $\phi_{\epsilon_{\Omega+1}}(0)$ plays the role of a super-ϵ_0, obtained by first going from ω to the first uncountable ordinal Ω and then coming back through the analogue $\epsilon_{\Omega+1}$ of ϵ_0. If we define Ω_{n+1} as the first ordinal of cardinality greater than Ω_n, then $\phi_{\epsilon_{\Omega_{n+1}+1}}(0)$ plays the role of a super-$\phi_{\epsilon_{\Omega_n+1}}(0)$. Girard proves (see Wainer [1989] for a recursion-theoretical proof) that

$$h_{\phi_{\epsilon_{\Omega_n+1}}(0)} \equiv_{\mathcal{E}} h_{\phi_{\epsilon_{\Omega_{n+1}+1}}(0)}.$$

If $\Omega_\omega = \lim_{n\to\infty} \Omega_n$ and $\phi_{\Omega_\omega}(0) = \lim_{n\to\infty} \phi_{\epsilon_{\Omega_n+1}}(0)$, then

$$h_{\phi_{\Omega_\omega}(0)} \equiv_{\mathcal{E}} f_{\phi_{\Omega_\omega}(0)},$$

i.e. $\phi_{\Omega_\omega}(0)$ *is the first meeting point of the Fast and Slow Growing Hierarchies*, and it constitutes a limit to the length of a natural extension of the Grzegorczyk Hierarchy into the transfinite. Indeed, the definition of $\{\mathcal{E}_\alpha\}_{\alpha<\gamma}$ reduces to the definition of h_α for $\alpha < \gamma$, which in turn reduces to the choice of the system of notations for α. To consider the hierarchy as natural, we must require that the system of notations for α is obtained before the definition of h_α, i.e. before stage α. Since the system of notations for α is faithfully represented by f_α, this amounts to requiring that $f_\alpha \in \bigcup_{\beta<\alpha} \mathcal{E}_\beta$. Since f_α is honest, i.e. computable in time elementary in f_α, f_α is elementary in any function dominating it. In particular, h_α is naturally defined only if it dominates f_α, i.e. only for $\alpha < \phi_{\Omega_\omega}(0)$. Thus $\mathcal{E}_{\phi_{\Omega_\omega}(0)}$ *is a maximal natural subclass of the recursive functions*, and as such it constitutes an appropriate stopping point of the present chapter.

æ

Chapter IX

Recursively Enumerable Sets

In the first part of this chapter we consider the classes of recursive and r.e. sets and study them from a set-theoretical point of view, as lattices under inclusion. In Section 1 we dispose of the **recursive sets** by characterizing their lattice in an algebraic way and completely solving all usual global questions about definability, homogeneity, and automorphisms. In Section 2 we study a number of natural classes of **recursively enumerable sets** defined in terms of lattice-theoretical properties of the principal filters generated by them. In Section 3 we turn to global questions for the lattice of r.e. sets and dispose of definability, homogeneity, and automorphisms, but leave open the algebraic characterization of the structure.

In the second part of the chapter we extend to r.e. sets (and partial recursive functions) the work done in Chapter VII for total recursive functions. More precisely, in Section 4 we develop a **complexity theory** for r.e. sets, while in Section 5 we indicate the changes needed to adapt the theory of **inductive inference** to the context of r.e. sets.

Since we deal with sets modulo finite sets, we introduce some **notation**:

1. $\boldsymbol{A} \subseteq^* \boldsymbol{B}$ means that A is almost contained in B, i.e. $A \subseteq B \cup F$ for some finite set F

2. $\boldsymbol{A} =^* \boldsymbol{B}$ means that A and B differ at most finitely, i.e. $A \subseteq^* B$ and $B \subseteq^* A$

3. $\boldsymbol{A^*}$ is the equivalence class of A w.r.t. $=^*$, i.e. $\{B : A =^* B\}$

4. for any class $\mathcal{C}$ of sets, $\boldsymbol{\mathcal{C}^*} = \{A^* : A \in \mathcal{C}\}$.

IX.1 Global Properties of Recursive Sets

We start by looking at the particularly well-behaved lattice of recursive sets.

Definition IX.1.1 **REC** *is the structure of recursive sets, ordered by inclusion. Similarly,* **REC**** *is the structure of recursive sets modulo finite sets, ordered by inclusion.*

Characterizations of the lattice of recursive sets

We already know (II.1.22) that the recursive sets form a countably infinite Boolean algebra. A small piece of additional information enables us to characterize completely both **REC** and **REC***. Since the proofs have nothing to do with Recursion Theory, we just sketch them here. For additional details, see Ershov [1980] or Monk and Bonnett [1989].

Theorem IX.1.2 Characterization of REC*. **REC*** *is characterized up to isomorphism (in a first-order way) by the property of being a countably infinite atomless Boolean algebra.*

Proof. Since we already know that **REC*** is a countably infinite Boolean algebra, we only have to prove that it is atomless. This follows from the fact that every infinite recursive set can be trivially split into two disjoint and infinite recursive sets.

It remains to prove that *if $\mathcal{A}$ and $\mathcal{B}$ are two countably infinite atomless Boolean algebras, they are isomorphic* (Cantor). The isomorphism $\phi : \mathcal{A} \to \mathcal{B}$ will be obtained as the limit of finite isomorphisms $\phi_n : \mathcal{A}_n \to \mathcal{B}_n$ between finite subalgebras of $\mathcal{A}$ and $\mathcal{B}$.

Since $\mathcal{A}$ and $\mathcal{B}$ are countable, we can fix enumerations of them. We start with $\mathcal{A}_0 = \mathcal{B}_0 = \{0, 1\}$, $\phi_0(0) = 0$ and $\phi_0(1) = 1$. Given $\phi_n : \mathcal{A}_n \to \mathcal{B}_n$, we proceed as follows:

- *n even*
  Let a be the smallest element of $\mathcal{A}$ (in the given enumeration) which is not yet in $\mathcal{A}_n$. We want to extend the isomorphism ϕ_n to a. Choose any element b in $\mathcal{B}$ which is in the same relationship with the elements of $\mathcal{B}_n$ as a is with the elements of $\mathcal{A}_n$ corresponding to them (via ϕ_n^{-1}). Such an element exists because $\mathcal{B}$ is an atomless Boolean algebra (e.g. $\mathcal{B}$ is dense – otherwise the difference of two elements with nothing in between would be an atom – and thus we can always pick up an element between any

two given elements). By letting $\phi(a) = b$, we preserve the order relation on the elements of $\mathcal{A}_n$ and $\mathcal{B}_n$ (by the choice of b). We thus let

$$\begin{aligned}\mathcal{A}_{n+1} &= \text{the finite subalgebra of } \mathcal{A} \text{ generated by } \mathcal{A}_n \cup \{a\}\\ \mathcal{B}_{n+1} &= \text{the finite subalgebra of } \mathcal{B} \text{ generated by } \mathcal{B}_n \cup \{b\},\end{aligned}$$

and define ϕ_{n+1} as the natural extension of ϕ_n induced by the additional condition $\phi_{n+1}(a) = b$.

- *n is odd*
  We proceed as above, with the roles of $\mathcal{A}$ and $\mathcal{B}$ interchanged. □

Notice that, in particular, **REC*** *is isomorphic to the Boolean algebra generated by the half-open intervals of the rationals.*

Theorem IX.1.3 Characterizations of REC. **REC** *is characterized:*

1. *up to elementary equivalence (in a first-order way) by the property of being a countably infinite atomic Boolean algebra*
2. *up to isomorphism (in a second-order way) by the property of being a countably infinite atomic Boolean algebra which induces, modulo the ideal generated by the atoms, a countably infinite atomless Boolean algebra.*

Proof. **REC** obviously satisfies the stated properties (the atoms being the singletons). It remains to prove:

1. *If $\mathcal{A}$ and $\mathcal{B}$ are countably infinite atomic Boolean algebras then $\mathcal{A} \equiv \mathcal{B}$, i.e. $\mathcal{A}$ and $\mathcal{B}$ are elementarily equivalent* (Tarski)
   We use the following standard criterion for elementary equivalence. Given two structures $\mathcal{A}$ and $\mathcal{B}$ with the same signature, and with distinguished constants (not necessarily the same for $\mathcal{A}$ and $\mathcal{B}$), let

   $$\mathcal{A} \equiv_0 \mathcal{B} \quad \Leftrightarrow \quad \text{the structures generated by the constants are isomorphic}$$

   and

   $$\begin{aligned}\mathcal{A} \equiv_{n+1} \mathcal{B} \quad \Leftrightarrow \quad & (\forall a \in \mathcal{A})(\exists b \in \mathcal{B})[(\mathcal{A}, a) \equiv_n (\mathcal{B}, b)] \ \wedge \\ & (\forall b \in \mathcal{B})(\exists a \in \mathcal{A})[(\mathcal{A}, a) \equiv_n (\mathcal{B}, b)].\end{aligned}$$

   Then
   $$(\forall n)(\mathcal{A} \equiv_n \mathcal{B}) \Rightarrow \mathcal{A} \equiv \mathcal{B}.$$

   This follows from the fact that if φ is a sentence with n quantifiers (in prenex normal form), then

   $$\mathcal{A} \equiv_n \mathcal{B} \Rightarrow (\mathcal{A} \models \varphi \Leftrightarrow \mathcal{B} \models \varphi).$$

   We prove this by induction.

- For $n = 0$, this is trivial. A formula φ with no quantifiers expresses a relation among constants, and if $\mathcal{A} \equiv_0 \mathcal{B}$ the structures generated by the constants are isomorphic, and hence they satisfy the same formulas.
- For $n+1$, say e.g. φ is $\exists x\psi$. If $\mathcal{A} \models \varphi$, then $(\mathcal{A}, a) \models \psi(a)$, for some $a \in \mathcal{A}$. If $\mathcal{A} \equiv_{n+1} \mathcal{B}$, then $(\mathcal{A}, a) \equiv_n (\mathcal{B}, b)$ for some b. Then we can apply the induction hypothesis to ψ, which has only n quantifiers. So $(\mathcal{B}, b) \models \psi(b)$ and $\mathcal{B} \models \varphi$. The converse is proved similarly.

Now let $\mathcal{A}$ and $\mathcal{B}$ be countably infinite atomic Boolean algebras. By the criterion just proved, to show that $\mathcal{A} \equiv \mathcal{B}$ it is enough to show that $\mathcal{A} \equiv_n \mathcal{B}$ for every n. For $n = 0$ this is automatic, since the structures generated by the constants are in both cases the two-element Boolean algebra. For $n > 0$, given e.g. $a \in \mathcal{A}$, we have three cases:

- If a is in the ideal generated by the atoms, it is uniquely determined by a finite number of atoms, and we can take a $b \in \mathcal{B}$ which is determined in the same way by a set of atoms with the same number of elements.
- If a is in the filter generated by the coatoms, we proceed similarly.
- Otherwise, we cannot simply take an element $b \in \mathcal{B}$ which is not in the ideal generated by the atoms nor in the filter generated by the coatoms, since such an element might not exist (for example, consider the Boolean algebra consisting of just the finite or cofinite sets of numbers). But since we only want $(\mathcal{A}, a) \equiv_n (\mathcal{B}, b)$ for a fixed n, and this amounts to a consideration of finite structures, it is enough to take $b \in \mathcal{B}$ such that the structures below it and above its complement are rich enough.

2. *If $\mathcal{A}$ and $\mathcal{B}$ are countably infinite atomic Boolean algebras inducing isomorphic Boolean algebras modulo the ideal generated by the atoms, they are isomorphic* (Vaught)
We only have to extend an isomorphism ψ between quotients (defined on equivalence classes $[a]$) to an isomorphism ϕ between $\mathcal{A}$ and $\mathcal{B}$ (defined on elements a), in such a way that ϕ induces ψ.

   Given any element a of $\mathcal{A}$, since we want ϕ to induce ψ, we must have

$$\phi(a) = b \Rightarrow \psi([a]) = [b],$$

   and thus $\phi(a)$ will be chosen in $\psi([a])$. If $\psi(c)$ has not yet been defined for any element c of $[a]$, we can choose as $\phi(a)$ any element of $\psi([a])$. Otherwise, suppose $\phi(c)$ has already been defined; since a and c belong to the same equivalence class, they differ only on finitely many atoms. We thus define ϕ on each of these atoms (for which it has not yet been defined), by sending them to appropriate atoms of $\mathcal{B}$; this is always possible, since $\mathcal{A}$ and $\mathcal{B}$ have countably many atoms. Then we choose as $\phi(a)$ an element of $\psi([a])$ having the same relation with $\phi(c)$ and the relevant atoms of $\mathcal{B}$, as a has with c and the relevant atoms of $\mathcal{A}$. $\square$

A reformulation of part 2 above is the following: **REC** *is characterized in* $\mathcal{P}(\omega)$*, up to isomorphism, by the property of being a countably infinite Boolean algebra containing all finite sets as well as some infinite and coinfinite set, and such that every infinite element can be split into two infinite elements.*

Exercise IX.1.4 *It is impossible to characterize* **REC** *up to isomorphism, in a first-order way.* (Hint: since any two countably infinite atomic Boolean algebras are elementarily equivalent, it is enough to exhibit two nonisomorphic countably infinite atomic Boolean algebras. For example, consider **REC** and the Boolean algebra of finite or cofinite sets: they are not isomorphic, because their quotients modulo the ideal generated by the atoms are not isomorphic.)

The complexity of the theory of recursive sets

The characterization results proved above allow us to determine the complexity of the theories of **REC** and **REC***.

Theorem IX.1.5 (Lachlan [1968b]) **REC** *and* **REC*** *are decidable.*

Proof. We have noted in Section III.10 that a consistent and complete formal system is decidable. Now the theories of **REC** and **REC*** are obviously complete, being theories of models. By IX.1.2 and IX.1.3, they are axiomatizable (as atomic and atomless Boolean algebras, respectively). $\square$

Note that, although both decidable, **REC** and **REC*** have different decision procedures, being not elementarily equivalent (e.g. one is atomic, and the other is not). Lachlan [1968b] exhibits direct decision procedures for them.

Homogeneity

We now look at intervals of **REC** and **REC***, and ask whether they resemble the whole structure.

Theorem IX.1.6 Homogeneity for REC and REC*.

1. *For any A and B in* **REC** *such that $A \subseteq B$, the segment*
$$[A, B] = \{X : X \in \mathbf{REC} \ \wedge \ A \subseteq X \subseteq B\}$$
is isomorphic to **REC** *if and only if $B - A$ is infinite.*

2. *For any A^* and B^* in* **REC*** *such that $A \subseteq^* B$, the segment*
$$[A^*, B^*] = \{X^* : X^* \in \mathbf{REC}^* \ \wedge \ A \subseteq^* X \subseteq^* B\}$$
is isomorphic to **REC*** *if and only if $A^* \neq B^*$.*

Proof. Clearly it is necessary that B differ infinitely from A. For sufficiency, if $B - A$ is an infinite recursive set, let $f : \omega \to (B - A)$ be a recursive function enumerating $B-A$ in order of magnitude (II.1.17). Then the map $C \mapsto A \cup f(C)$ induces the required isomorphism in both parts 1 and 2. $\Box$

Exercise IX.1.7 *Every countable partial ordering is embeddable in any nontrivial segment of* **REC***. (Goetze [1976]) (Hint: by homogeneity, it is enough to prove the result for **REC*** itself. This comes from the fact that **REC*** is a countably infinite atomless Boolean algebra.)

One can also look at 'external' homogeneity questions, by considering the Boolean algebra of sets recursive in a given set A. It is an immediate consequence of the characterization theorems IX.1.2 and IX.1.3 that *for any set A, the Boolean algebra* **REC**^{A} *of sets recursive in A is isomorphic to* **REC**, and similarly for the algebras modulo finite sets.

Automorphisms

Since there are only $2^{\aleph_0}$ maps of a countably infinite structure into itself, the following result is the best possible and completely solves the question of the number of automorphisms (i.e. one-one onto maps preserving the order) for **REC** and **REC***.

Theorem IX.1.8 Automorphisms of REC and REC* (Lachlan)

1. **REC** *and* **REC*** *both have $2^{\aleph_0}$ automorphisms, all of them induced by permutations of ω.*

2. *Every automorphism of* **REC** *induces an automorphism of* **REC****, and every automorphism of* **REC*** *is induced by an automorphism of* **REC**.

Proof. The existence of $2^{\aleph_0}$ automorphisms of $\mathbf{REC}^*$ follows from the proof of IX.3.18, which only requires choosing nontrivial permutations of given recursive sets R_n. This can be obtained by actually splitting each R_n into two infinite recursive parts, and interchanging them. Thus the $2^{\aleph_0}$ automorphisms of $\boldsymbol{\mathcal{E}^*}$ built in IX.3.18 differ on the recursive sets.

The argument of IX.1.3.2 actually shows that every automorphism of $\mathbf{REC}^*$ is induced by an automorphism of $\mathbf{REC}$, and hence there are $2^{\aleph_0}$ automorphisms of $\mathbf{REC}$ as well.

It is immediate that every automorphism of $\mathbf{REC}$ is induced by a permutation of ω, since an automorphism is completely determined by its behavior on singletons, which are sent to singletons. An automorphism ϕ is thus induced by the permutation

$$p(x) = \text{ the only element of } \phi(\{x\}).$$

Then every automorphism of $\mathbf{REC}^*$ is also so induced.

Finally, it is trivial to note that every automorphism of $\mathbf{REC}$ induces an automorphism of $\mathbf{REC}^*$. $\square$

The connections between automorphisms of $\mathbf{REC}^*$ and $\boldsymbol{\mathcal{E}^*}$ (the lattice of r.e. sets modulo finite sets) is discussed in IX.3.20.

Absolute definability

The next result shows that nothing nice can be lattice-theoretically defined in either $\mathbf{REC}$ or $\mathbf{REC}^*$.

Theorem IX.1.9 Non-Definability in REC and REC*.

1. *In* $\mathbf{REC}$ *there is no nontrivial definable ideal, filter, finite or cofinite set of elements.*
2. *In* $\mathbf{REC}^*$ *there is no nontrivial definable set.*

Proof. We first prove part 2. Given any two infinite coinfinite recursive sets there is a recursive permutation of ω (defined by a dovetailed matching of the sets and their complements in increasing order) which interchanges the two sets. This induces an automorphism of $\mathbf{REC}^*$ which interchanges any two given distinct elements of $\mathbf{REC}^*$ different from 0 and 1. Now, given any nontrivial subset $\mathcal{A}$ of $\mathbf{REC}^*$, take an element different from 0 and 1 in it, and send it via an automorphism to an element different from 0 and 1 and not in $\mathcal{A}$: then $\mathcal{A}$ is not closed under automorphisms, and thus is not definable.

We now prove part 1. Given a finite subset $\mathcal{A}$ of $\mathbf{REC}$, if every element of $\mathcal{A}$ is infinite and coinfinite then the proof above shows that $\mathcal{A}$ is not definable.

If $\mathcal{A}$ contains a finite set, it is trivial to define an automorphism of **REC** which sends this finite set into a finite set not belonging to $\mathcal{A}$ (since the latter is finite), and thus $\mathcal{A}$ is not definable. Similarly if $\mathcal{A}$ contains a cofinite set.

We now turn to ideals (the case of filters being symmetric). Suppose $\mathcal{I}$ is a definable nontrivial ideal, and let $\mathcal{F}$ be the ideal consisting of the finite sets. We obtain a contradiction by showing that $\mathcal{I}$ is not definable in **REC**, in two steps:

- $\mathcal{I} = \mathcal{F}$ (Lerman [1976])
  Since $\mathcal{I}$ is nontrivial, it has a nonempty member A. If $x \in A$, then $\{x\} \subseteq A$, and $\{x\} \in \mathcal{I}$ because $\mathcal{I}$ is an ideal. For any y, $\{x\}$ and $\{y\}$ are automorphic and so $\{y\} \in \mathcal{I}$, because $\mathcal{I}$ is definable and hence closed under automorphisms. Then all singletons are in $\mathcal{I}$, and $\mathcal{F} \subseteq \mathcal{I}$ because $\mathcal{I}$ is closed under finite unions.

  Now suppose that $\mathcal{I}$ has an infinite recursive member C: there is an infinite, coinfinite recursive set B contained in it, and $B \in \mathcal{I}$ because $\mathcal{I}$ is downward closed. But B and $\overline{B}$ are automorphic by the proof above, and thus $\overline{B} \in \mathcal{I}$ because $\mathcal{I}$ is definable. Thus $\omega = B \cup \overline{B} \in \mathcal{I}$, and then $\mathcal{I}$ is trivial, which is a contradiction. Thus $\mathcal{I} \subseteq \mathcal{F}$.

- *$\mathcal{F}$ is not definable in* **REC**
  Suppose there is a formula φ that defines $\mathcal{F}$ in **REC**. By the Compactness Theorem, there is a countably infinite Boolean algebra $\mathcal{A}$ extending **REC** and elementarily equivalent to it, and such that there is in $\mathcal{A}$ an infinite element a satisfying φ (where infinite means that, for every n, there are n distinct atoms below a). Since $\mathbf{REC}^*$ is atomless, so is $\mathcal{A}^*$; thus $\mathbf{REC}^*$ and $\mathcal{A}^*$ are isomorphic by the proof of IX.1.2, and hence so are **REC** and $\mathcal{A}$, by the proof of IX.1.3.2. But the isomorphism should send a into an infinite set (because the fact of being above n distinct atoms must be preserved, for every n) satisfying φ, which is a contradiction. $\Box$

Part 1 cannot be substantially improved because *in* **REC** *there are infinitely many definable (infinite) sets*. For example, for each n, the class of sets with at least n elements (definable by the property of being above n distinct atoms).

Ultrafilters and models of fragments of Arithmetic ⋆

The following notion is patterned after the notion of cohesiveness (III.4.13).

Definition IX.1.10 *A set A is* ***r*-cohesive** *or* **recursively indecomposable** *if it is infinite, and cannot be split into two infinite parts by a recursive set. That is, if B is recursive, then either $B \cap A$ or $\overline{B} \cap A$ is finite.*

Recall that an *ultrafilter* $\mathcal{U}$ in a Boolean algebra $\mathcal{A}$ of sets is a maximal filter or, equivalently, a filter such that, for every A in $\mathcal{A}$, either A or $\overline{A}$ are in $\mathcal{U}$.

The next result characterizes the nonprincipal ultrafilters of $\mathbf{REC}^*$, as the intersections of $\mathbf{REC}^*$ with the principal filters of $\mathcal{P}(\omega)$ (modulo the finite sets) generated by recursively indecomposable sets.

Proposition IX.1.11 (Nerode [1966]) *In* $\mathbf{REC}^*$ *the nonprincipal ultrafilters are exactly those of the form*

$$\mathcal{U} \;=\; \{X^* : A \subseteq^* X \;\wedge\; X \text{ recursive}\},$$

where A is an r-cohesive set.

Proof. If $\mathcal{U}$ is defined as above, then it is trivially a filter. Moreover, it is an ultrafilter because, given any recursive set X, one of $X \cap A$ and $\overline{X} \cap A$ is finite (by r-cohesiveness of A). In the first case, $\overline{X}$ differs finitely from a recursive set containing A, and thus $\overline{X} \in \mathcal{U}$. Similarly, in the second case $X \in \mathcal{U}$.

Conversely, if $\mathcal{U}$ is a nonprincipal ultrafilter, then choose a list $\{X_n\}_{n\in\omega}$ of one representative for each equivalence class of $\mathcal{U}$, and define a set A by enumerating it as follows:

$$A(n) = \mu x.(x > A(n-1) \;\wedge\; x \in X_0 \cap \cdots \cap X_n).$$

A is infinite because each $A(n)$ is defined: indeed, each finite intersection of the X_n's is infinite, otherwise $\mathcal{U}$ would be principal. Moreover, A is almost contained in each X_n, since every element from the n-th one is in A by definition: then the filter

$$\{X^* : A \subseteq^* X \;\wedge\; X \text{ recursive}\}$$

contains $\mathcal{U}$, and hence it is equal to it by maximality (because $\mathcal{U}$ is an ultrafilter). Finally, A is r-cohesive because, given any recursive set X, $X^* \in \mathcal{U}$ or $\overline{X^*} \in \mathcal{U}$ by the definition of an ultrafilter; thus one of X and $\overline{X}$ contains A except for a finite difference, and hence no recursive set can split A into two infinite parts. $\square$

Ultrafilters are interesting because they provide a paradigm for constructing models of (fragments of) Arithmetic, as follows. Given $\mathcal{U}$, identify functions on ω that agree on an element of $\mathcal{U}$:

$$f \equiv_{\mathcal{U}} g \;\Leftrightarrow\; \{x : f(x) = g(x)\} \in \mathcal{U},$$

the idea being that the elements of a filter are 'big', and thus f and g are 'almost equal'. This provides an equivalence relation on ω^ω with respect to which we can factor, thus obtaining a structure $\omega^\omega/\mathcal{U}$ whose elements are equivalence classes $[f]$. The map

$$n \longmapsto [C_n],$$

where C_n is the constant function with value n, provides a canonical embedding of ω into $\omega^\omega/\mathcal{U}$, which is one-one because two C_n's either agree everywhere, or disagree everywhere.

Arithmetical functions and relations naturally induce functions and relations on $\omega^\omega/\mathcal{U}$. The former by first inducing functions on ω^ω in a pointwise manner, and then factoring w.r.t. the equivalence relation. The latter by asking, as in the case of equality, that the set of components on which the relation holds on ω is an element of $\mathcal{U}$. For example,

$$\begin{aligned} [f\,] + [g\,] &= [f+g\,] \\ [f\,] < [g\,] &\Leftrightarrow \{x : f(x) < g(x)\} \in \mathcal{U}. \end{aligned}$$

Thus every arithmetical formula φ can be interpreted on $\omega^\omega/\mathcal{U}$, and the main fact is that

$$\omega^\omega/\mathcal{U} \models \varphi([f_1], \dots, [f_n]) \Leftrightarrow \{x : \omega \models \varphi(f_1(x), \dots, f_n(x))\} \in \mathcal{U}.$$

This is easily proved by induction on φ, using the fact that $\mathcal{U}$ is an ultrafilter to deal with negations, and the filter properties in the other cases.

In particular, *$\omega^\omega/\mathcal{U}$ is a model of Arithmetic* (with the natural number n interpreted as $[C_n]$) because

$$\begin{aligned} \omega^\omega/\mathcal{U} \models \varphi([C_{x_1}], \dots, [C_{x_n}]) &\Leftrightarrow \{x : \omega \models \varphi(C_{x_1}(x), \dots, C_{x_n}(x))\} \in \mathcal{U} \\ &\Leftrightarrow \{x : \omega \models \varphi(x_1, \dots, x_n)\} \in \mathcal{U} \\ &\Leftrightarrow \omega \models \varphi(x_1, \dots, x_n), \end{aligned}$$

since by definition $\emptyset \notin \mathcal{U}$.

If $\mathcal{U}$ is a principal ultrafilter, then $\omega^\omega/\mathcal{U}$ is isomorphic to the standard model ω, because the canonical embedding is onto in this case. Indeed, given $[f\,]$, consider the sets $X_n = \{x : f(x) = n\}$. The X_n are disjoint, and one of them must be in $\mathcal{U}$ (otherwise their complements are, since $\mathcal{U}$ is an ultrafilter, and hence their intersection is, since $\mathcal{U}$ is principal; but the intersection is empty, while $\mathcal{U}$ is nontrivial). If $X_n \in \mathcal{U}$, then $[f\,] = [C_n]$.

On the other hand, *if $\mathcal{U}$ is nonprincipal, then $\omega^\omega/\mathcal{U}$ is a nonstandard model.* For example, the identity function (identified with the sequence of its values) is the beginning of an infinitely descending sequence:

$$\langle 0,1,2,3,4,\dots\rangle > \langle 0,0,1,2,3,\dots\rangle > \langle 0,0,0,1,2,\dots\rangle > \cdots$$

The inequalities hold because each function in the sequence differs from the following one on a cofinite set, and if this were not in $\mathcal{U}$, then its complement would be. But an ultrafilter can contain a finite set only if it is principal.

This procedure for building (nonstandard) models of Arithmetic was introduced by Skolem [1934], who actually carried it a step further. By considering only arithmetical functions and factoring them by a nonprincipal ultrafilter on the arithmetical sets, he obtained a *countable nonstandard model of Arithmetic*. The main reason why arithmetical functions are enough is that they skolemize every arithmetical formula. A relativization of IX.1.11 provides a description of the nonprincipal ultrafilters of arithmetical sets (modulo finite sets), as the intersections of the latter with the principal ultrafilters of $\mathcal{P}(\omega)$ generated by arithmetically indecomposable sets. In particular, if $\mathcal{U}$ is generated by such a set A, then

$$f \equiv_{\mathcal{U}} g \;\Leftrightarrow\; f \text{ and } g \text{ agree on } A \text{ with at most finitely many exceptions.}$$

One can further miniaturize Skolem's result, and consider the recursive functions factored by a nonprincipal ultrafilter on the recursive sets obtained as in IX.1.11. Since recursive functions only skolemize Σ_2^0 sentences (II.1.13), one then obtains *nonstandard models of two-quantifier Arithmetic* (Hirschfeld [1975]). Feferman, Scott and Tennenbaum [1959] have proved that *one never obtains models of First-Order Arithmetic* when factoring the recursive functions by a nonprincipal ultrafilter. To see this, consider the formula (true in ω)

$$(\forall x)(\exists s)(\forall e \leq x)[\varphi_e(x)\!\downarrow \;\Rightarrow\; \varphi_{e,s}(x)\!\downarrow].$$

When interpreting it in the structure of recursive sets modulo a nonprincipal ultrafilter $\mathcal{U}$, we can take $x = [\mathcal{I}]$ where $\mathcal{I}$ is the identity function. Then we should have a recursive function g (playing the role of s) such that, whenever $[f] \leq [\mathcal{I}]$, i.e. $f(x) \leq x$ on a set in $\mathcal{U}$ (with f playing the role of e), then, by the pointwise interpretation of relations,

$$\varphi_{f(x)}(x)\!\downarrow \;\Rightarrow\; \varphi_{f(x),g(x)}(x)\!\downarrow$$

holds on a set in $\mathcal{U}$. By taking f as the constant function with value e, for e such that $g \simeq \varphi_e$, we have $[f] \leq [\mathcal{I}]$. Hence

$$\varphi_e(x)\!\downarrow \;\Rightarrow\; \varphi_{e,g(x)}(x)\!\downarrow$$

should hold on a set in $\mathcal{U}$. But this says that when $\varphi_e(x)$ converges, it does so in less than $g(x) = \varphi_e(x)$ steps for a large set of x (in particular an infinite one, if $\mathcal{U}$ is not principal), while a recursive function can be computed in a number of steps smaller than its values only for finitely many arguments (see II.1.4.2).

Lerman [1970a] shows that, contrary to the arithmetical case, where all models thus obtained are elementarily equivalent because they are models of First-Order Arithmetic, *in the recursive case there are different ultrafilters that generate nonelementarily equivalent models*.

IX.2 Local Properties of R.E. Sets

We turn now to the lattice of r.e. sets, defined as follows.

Definition IX.2.1 (Myhill [1956]) $\boldsymbol{\mathcal{E}}$ *is the structures of r.e. sets, ordered by inclusion. Similarly,* $\boldsymbol{\mathcal{E}^*}$ *is the structure of r.e. sets modulo finite sets, ordered by inclusion.*

We already know (II.1.21) that the r.e. sets form a distributive lattice with smallest and greatest element, such that the recursive sets are the only complemented elements. These properties are far from characterizing the structures of $\mathcal{E}$ and $\mathcal{E}^*$, and nothing in the style of IX.1.2 and IX.1.3 is known for them. We now start a study of the local structure of $\mathcal{E}$ and $\mathcal{E}^*$, and will return to global properties in the next section.

We concentrate in this section on principal filters of $\mathcal{E}$, defined as follows:

Definition IX.2.2 $\boldsymbol{\mathcal{L}(A)}$ *is the* **principal filter** *of* $\mathcal{E}$ *generated by the r.e. set A, i.e.*

$$\mathcal{L}(A) = \{B : B \in \mathcal{E} \ \wedge \ A \subseteq B\}.$$

$\boldsymbol{\mathcal{L}^*(A)}$ *is the principal filter of* $\mathcal{E}^*$ *induced by* $\mathcal{L}(A)$.

The structure of principal filters $\mathcal{L}^*(A)$ can be highly varied, and the following important cases are considered in the following:

$\mathcal{L}^*(A) \cong \{0,1\}$	$\Leftrightarrow$	A maximal
$\mathcal{L}^*(A)$ finite	$\Leftrightarrow$	A quasimaximal (IX.2.22)
$\mathcal{L}^*(A)$ Boolean algebra	$\Leftrightarrow$	A hyperhypersimple (IX.2.10)
$\mathcal{L}^*(A)$ with no complements	$\Leftrightarrow$	A r-maximal (IX.2.12)
$\mathcal{L}^*(A)$ dense	$\Leftrightarrow$	A without maximal supersets (IX.2.26).

Splitting theorems

We have seen in Section 1 that the (trivial) fact that every infinite recursive set can be split into two infinite recursive sets is crucial for the characterization of the structure **REC** up to isomorphism. We now look briefly at splitting properties for r.e. sets, and refer to Downey and Stob [1993] for a comprehensive survey.

It is useful here to consider the r.e. sets not only extensionally but dynamically, according to the order in which their elements are generated. The following notions have already been used informally, e.g. in the proof of II.1.23.

Definition IX.2.3 $\boldsymbol{\mathcal{W}_e \setminus \mathcal{W}_i}$ *is the set of elements generated in* $\mathcal{W}_e$ *before being generated in* $\mathcal{W}_i$*:*

$$\mathcal{W}_e \setminus \mathcal{W}_i = \{x : (\exists s)(x \in \mathcal{W}_{e,s} - \mathcal{W}_{i,s})\}.$$

$\boldsymbol{\mathcal{W}_e \searrow \mathcal{W}_i}$ *is the set of elements of* $\mathcal{W}_e \setminus \mathcal{W}_i$ *that later turn out to be in* $\mathcal{W}_i$*:*

$$\mathcal{W}_e \searrow \mathcal{W}_i = \mathcal{W}_i \cap (\mathcal{W}_e \setminus \mathcal{W}_i).$$

$\mathcal{W}_e \setminus \mathcal{W}_i$ is clearly a computational complexity notion, and we could have defined more generally (in the notation of Chapter VII):

$$x \in (\mathcal{W}_e \setminus \mathcal{W}_i) \;\Leftrightarrow\; \Phi_e(x) < \Phi_i(x).$$

Since an element of $\mathcal{W}_e \setminus \mathcal{W}_i$ may eventually turn out to be in $\mathcal{W}_i$ (and hence be in $\mathcal{W}_e \searrow \mathcal{W}_i$) or not (and hence be in $\mathcal{W}_e - \mathcal{W}_i$), we have

$$\mathcal{W}_e \setminus \mathcal{W}_i = (\mathcal{W}_e \searrow \mathcal{W}_i) \cup (\mathcal{W}_e - \mathcal{W}_i).$$

We also use the notation $A \setminus B$ and $A \searrow B$ for r.e. sets A and B whose enumerations are understood.

Theorem IX.2.4 Friedberg-Muchnik Splitting Theorem (Friedberg [1958], Muchnik [1958]) *Every noncomplemented element of* $\boldsymbol{\mathcal{E}}$ *can be split into two noncomplemented elements. That is, for any nonrecursive r.e. set* A*, there are disjoint nonrecursive r.e. sets* B *and* C *such that* $A = B \cup C$*.*

Proof. We effectively enumerate A, and whenever an element is generated in it we put it in exactly one of B or C, thus making B and C disjoint with union A. The construction will be recursive, and thus B and C will be r.e.

The requirements to make B and C nonrecursive are the following:

$$\begin{array}{lcl} P_e^B & : & \overline{B} \subseteq \mathcal{W}_e \;\Rightarrow\; \mathcal{W}_e \cap B \neq \emptyset \\ P_e^C & : & \overline{C} \subseteq \mathcal{W}_e \;\Rightarrow\; \mathcal{W}_e \cap C \neq \emptyset. \end{array}$$

We choose these because the symmetric ones, such as

$$\mathcal{W}_e \subseteq \overline{B} \;\Rightarrow\; \overline{\mathcal{W}}_e \cap \overline{B} \neq \emptyset,$$

have a negative effect, and are thus not suitable for the construction of r.e. sets.

Since the construction will make B and C subsets of A, if $\overline{B} \subseteq \mathcal{W}_e$ or $\overline{C} \subseteq \mathcal{W}_e$, then $\overline{A} \subseteq \mathcal{W}_e$. Moreover, since in this case

$$\overline{A} \cup (\mathcal{W}_e \searrow A) = \mathcal{W}_e \setminus A,$$

and $\mathcal{W}_e \setminus A$ is r.e. by definition, if $\mathcal{W}_e \searrow A$ were finite, then $\overline{A}$ would be r.e. But A is not recursive, and thus $\mathcal{W}_e \searrow A$ is infinite; this means that infinitely many elements are enumerated in $\mathcal{W}_e$ before they are enumerated in A, and we can then put one in B and one in C to satisfy the requirements.

The construction is now as follows. Fix a one-one recursive enumeration f of A, which exists because A is infinite, being nonrecursive. Start with $B_0 = C_0 = \emptyset$. At stage $s+1$, see if there is $e \leq s$ such that one of P_e^B and P_e^C has not yet been satisfied and can be satisfied now, i.e.

$$f(s) \in \mathcal{W}_{e,s} \text{ and } (\mathcal{W}_{e,s} \cap B_s = \emptyset \text{ or } \mathcal{W}_{e,s} \cap C_s = \emptyset).$$

If such an e exists, choose the smallest one and put $f(s)$ into B if $\mathcal{W}_{e,s} \cap B_s = \emptyset$, and into C otherwise. Otherwise put, for example, $f(s)$ into B.

By the discussion above, every requirement is eventually satisfied. $\square$

The proof is a priority argument with no injuries, with the following priority ordering:

$$P_0^B > P_0^C > P_1^B > P_1^C > \cdots$$

Although the requirements are similar to those for the construction of simple sets, we certainly cannot make B or C simple, since each is contained in the other's complement.

Exercise IX.2.5 *Every r.e. nonrecursive set is the disjoint union of two recursively inseparable r.e. sets.* (Yates) (Hint: let R be a recursive set such that $B \subseteq R$ and $C \subseteq \overline{R}$, and $R = \mathcal{W}_e$. Let s_0 be a stage after which all requirements of priority higher than P_e^C that will ever be satisfied have been satisfied. Since $C \subseteq \overline{R}$, $(\forall s \geq s_0)(C_s \cap \mathcal{W}_{e,s} = \emptyset)$. But since $\mathcal{W}_e \searrow A$ is infinite because B is not recursive, the construction gives $C \cap \mathcal{W}_e \neq \emptyset$, which is a contradiction.)

Exercises IX.2.6 Promptly simple sets and the splitting property. A set A is **promptly simple** (Maass [1982]) if it is r.e. and coinfinite, and for some nondecreasing recursive function g and some enumeration $\{A_s\}_{s\in\omega}$ of A:

$$\mathcal{W}_e \text{ infinite} \Rightarrow \exists s \exists x[x \in (\mathcal{W}_{e,s+1} - \mathcal{W}_{e,s}) \wedge x \in A_{g(s)}],$$

i.e. not only does some element of every infinite r.e. set $\mathcal{W}_e$ go into A, but it does so, at the latest, shortly after showing up in $\mathcal{W}_e$. This is a computational complexity notion which, if $A = \mathcal{W}_i$, could be defined as

$$\mathcal{W}_e \text{ infinite} \Rightarrow \exists x[x \in (\mathcal{W}_e \cap A) \wedge \Phi_i(x) \leq g(\Phi_e(x))].$$

A set S has the **splitting property** (Maass, Shore and Stob [1981]) if it is r.e. and every r.e. nonrecursive set A can be split into two nonrecursive r.e. sets B and C such that $B \subseteq S$.

a) *Post's simple set* (III.2.11) *is promptly simple.*

b) *There are low promptly simple sets.* (Maass [1982]) (Hint: satisfy the requirements

$$\begin{array}{lcl} P_e & : & \mathcal{W}_e \text{ infinite } \Rightarrow \exists x \exists s [x \in (\mathcal{W}_{e,s+1} - \mathcal{W}_{e,s}) \cap A_{s+1}] \\ N_e & : & \{e\}_s^{A_s}(e)\downarrow \text{ infinitely often } \Rightarrow \{e\}^A(e)\downarrow . \end{array}$$

See X.3.3 for a discussion of N_e.)

c) *A set having the splitting property is promptly simple.*

d) *Every promptly simple set has the splitting property.* (Maass, Shore and Stob [1981]) (Hint: given a set S promptly simple w.r.t. the recursive function g, and a nonrecursive r.e. set A, modify the construction in IX.2.4 by putting $f(s)$ into B at stage $s+1$ if and only if $f(s) \in S_{g(s)}$ and $\mathcal{W}_{e,s} \cap B_s = \emptyset$, so that $B \subseteq S$. This works because of prompt simplicity.)

e) *Every hyperhypersimple set has the splitting property.* (Hint: otherwise one obtains an infinite array of disjoint r.e. sets intersecting $\overline{A}$ by repeated splitting.)

f) *The r.e. sets having the splitting property form a filter in $\mathcal{E}$.* (Hint: for closure under intersection, let S_0 and S_1 be given. Given A, split it and obtain $A_0 \subseteq S_0$ such that $A_0 \cup A_1 = A$. Then split A_0.)

Maass [1982] proved that the notion of *prompt simplicity is recursively invariant*, while Maass, Shore and Stob [1981] proved that *promptly simple or cofinite sets form a filter in $\mathcal{E}$.* For more on promptly simple sets, see also Ambos-Spies, Jockusch, Shore and Soare [1984] and Maass [1985].

Exercises IX.2.7 Nowhere simple sets. (Shore [1978a]) A set A is **nowhere simple** if it is r.e. and coinfinite, and for every r.e. set B such that $B \cap \overline{A}$ is infinite there is an infinite r.e. set C contained in $B \cap \overline{A}$. Thus not only does $\overline{A}$ contain an infinite r.e. set, but so does every infinite part of it determined by an r.e. set.

a) *There exists a nowhere simple set.* (Hint: construct r.e. sets A and C_e satisfying the requirements

$$\begin{array}{lcl} P_e & : & \overline{A} \neq \mathcal{W}_e \\ N_{e,n} & : & \mathcal{W}_e \cap \overline{A} \text{ infinite } \Rightarrow |C_e| \geq n. \end{array}$$

P_e is satisfied as usual. N_e is satisfied by keeping elements of $\mathcal{W}_e$ out of A. At stage $s+1$, if for some new x we discover that $x \in \mathcal{W}_{e,s} \cap \overline{A}_s$, we keep x out of A except to satisfy P_i for $i \leq e+n$, where n is the number of elements already kept out of A to satisfy N_e.)

b) *The nowhere simple sets generate $\mathcal{E}$.* (Hint: show that every r.e. set can be split into two disjoint nowhere simple sets. Given A put every element of it into one of B and C, and satisfy the requirements for nowhere simplicity of both as in IX.2.4. See also part c).)

c) *The sets B and C of IX.2.4 are automatically nowhere simple.* (Downey and Stob [1993]) (Hint: given $\mathcal{W}_e$ intersecting $\overline{B}$ infinitely often, consider $\mathcal{W}_e \cap C$. If it is infinite, there is nothing to do. Otherwise, the set $\mathcal{W}_e \cap \overline{A}$ is infinite; if it were not r.e., the construction would make it intersect with B and C.)

For more on nowhere simple sets see Miller and Remmel [1984] and Downey and Stob [1993].

The Friedberg-Muchnik Splitting Theorem showed that, for any r.e. set A such that $\overline{A}$ is not r.e., there are disjoint r.e. sets B and C such that

1. $A = B \cup C$
2. $\overline{B}$ and $\overline{C}$ are not r.e.

We now extend this from $\boldsymbol{\mathcal{E}} = \mathcal{L}(\emptyset)$ to any of its principal filters.

Theorem IX.2.8 Owings Splitting Theorem (Owings [1967], Lachlan) *For any r.e. set D, every noncomplemented element of $\mathcal{L}(D)$ can be split into two noncomplemented elements. That is, given an r.e. set A such that $\overline{A} \cup D$ is not r.e., there are disjoint r.e. sets B and C such that*

1. $A = B \cup C$

2. $\overline{B} \cup D$ *and* $\overline{C} \cup D$ *are not r.e.*

Proof. As in IX.2.4, we put something into B or C only if it comes out in A. The other requirements are now:

$$\begin{aligned} P_e^B &: \overline{B} \cup D \subseteq \mathcal{W}_e \Rightarrow \mathcal{W}_e \cap B \cap \overline{D} \neq \emptyset \\ P_e^C &: \overline{C} \cup D \subseteq \mathcal{W}_e \Rightarrow \mathcal{W}_e \cap C \cap \overline{D} \neq \emptyset. \end{aligned}$$

Exactly as in IX.2.4 we can show that, under the hypotheses of the requirements, $(\mathcal{W}_e \searrow A) \cap \overline{D}$ is infinite because $\mathcal{W}_e \supseteq \overline{A} \cup D$.

The intuitive strategy to satisfy P_e^B would thus be, for example, to wait for s such that $f(s) \in \mathcal{W}_{e,s} \cap \overline{D}_s$, and put $f(s)$ into B if $\mathcal{W}_{e,s} \cap B_s \cap \overline{D}_s = \emptyset$. There are however the following problems:

- even if $f(s) \in \mathcal{W}_{e,s} \cap \overline{D}_s$, we have no control over D: if $f(s) \in D$, P_e^B is later injured
- it is not enough to wait for an injury and pick up a new witness, since we are not sure that we will ever hit an element in $\overline{D}$ (i.e. P_e^B might be injured infinitely often and not be satisfied).

The strategy is thus modified as follows. When P_e^B gets highest priority, we keep on putting elements of $\mathcal{W}_e$ into B as witnesses. Sooner or later we will then hit an element of $\overline{D}$. But since we also want to satisfy the other requirements, we cannot put too many elements in B for the sake of P_e^B. We thus define a function $g^B(e,s)$ that goes to infinity if and only if $\mathcal{W}_e \cap B \cap \overline{D} = \emptyset$ (i.e. if P_e^B is not satisfied), and use it to bound the elements we put into B. We then reason as before, and prove that sooner or later we do hit an element of $\overline{D}$.

The definition of g^B is as follows: $g^B(e,s)$ is equal to s if

$$\mathcal{W}_{e,s} \cap B_s \cap \overline{D}_s = \emptyset$$

or

$$\mu x.\, x \in (\mathcal{W}_{e,s} \cap B_s \cap \overline{D}_s) \neq \mu x.\, x \in (\mathcal{W}_{e,s-1} \cap B_{s-1} \cap \overline{D}_{s-1}),$$

and equal to $g^B(e,s-1)$ otherwise.

Note that $g^B(e,s) = s$ (i.e. it places no restrictions, since if $x \in \mathcal{W}_{e,s}$, then $x \leq s$ by definition) when P_e^B looks trivially unsatisfied ($\mathcal{W}_{e,s} \cap B_s \cap \overline{D}_s = \emptyset$), or the smallest element of the approximation of $\mathcal{W}_e \cap B \cap \overline{D}$ has changed. Then

$$\mathcal{W}_e \cap B \cap \overline{D} \neq \emptyset \;\Leftrightarrow\; \lim_{s\to\infty} g^B(e,s) < \infty.$$

Indeed, take the smallest element of $\mathcal{W}_e \cap B \cap \overline{D}$ and wait for a stage s after which that element is in $\mathcal{W}_{e,s} \cap B_s$, and all the smaller elements have settled; g^B does not change after s (for this e). Conversely, if $\lim_{s\to\infty} g^B(e,s) < \infty$ then, from a certain stage on, the smallest element of $\mathcal{W}_{e,s} \cap B_s \cap \overline{D}_s$ does not change, hence it witnesses that $\mathcal{W}_e \cap B \cap \overline{D} \neq \emptyset$.

Note that it would not be enough to have $\mathcal{W}_{e,s} \cap B_s \cap \overline{D}_s \neq \emptyset$ for almost every stage s, since the witnesses for nonemptyness might always be different.

The construction is now as follows. Fix a one-one recursive enumeration f of A, which exists because A is infinite, since $\overline{A} \cup D$ is not r.e. Start with $B_0 = C_0 = \emptyset$. At stage $s+1$, see if there is $e \leq s$ such that

$$f(s) \in \mathcal{W}_{e,s} \;\wedge\; [f(s) \leq g^B(e,s) \text{ or } f(s) \leq g^C(e,s)].$$

If so, choose the smallest one and put $f(s)$ into B if $f(s) \leq g^B(e,s)$, and into C otherwise. Otherwise put, for example, $f(s)$ into B.

If $\overline{A} \cup D \subseteq \mathcal{W}_e$ then consider the following analogue of $\mathcal{W}_e \setminus A$:

$$S = \{x : (\exists s)[x \in \mathcal{W}_{e,s} \;\wedge\; x \notin A_s \;\wedge\; x \leq g^B(e,s)]\}.$$

S is r.e. If $\lim_{s\to\infty} g^B(e,s) = \infty$, then $S \supseteq \overline{A}$. So $S \cap A \cap \overline{D}$ is infinite (since $\overline{A} \cup D$ is not r.e.). If P_e^B is the least unsatisfied requirement, we now show that we can satisfy it, contradicting the fact that $\lim_{s\to\infty} g^B(e,s) = \infty$. Consider $x \in S \cap A \cap \overline{D}$; by definition

$$x \in \mathcal{W}_{e,s} \;\wedge\; x \notin A_s \;\wedge\; x \leq g^B(e,s)$$

for some s. When x shows up in A at some later stage $t+1$ with $t \geq s$, we put it into B because $x \leq g^B(e,s) \leq g^B(e,t)$ by monotonicity of g^B. $\square$

In the following exercises we consider extensions of the Owings Splitting Theorem involving Turing degrees (proved by the priority method introduced in Chapter X).

Exercises IX.2.9 Extensions of the Owings Splitting Theorem. a) *In IX.2.8, B and C can be taken to be of incomparable Turing degree.* (Morley and Soare [1975]) (Hint: use the Sacks agreement method to satisfy the following sufficient requirements:

$$R_e^B \;:\; \mathcal{W}_e^B \cap \overline{D} \neq \overline{A} \cap \overline{D} \qquad \text{and} \qquad \mathcal{W}_e^C \cap \overline{D} \neq \overline{A} \cap \overline{D}.$$

In the construction, approximate $\overline{D}$ by a recursive function, which is possible because D is r.e., and look at the agreement between $\mathcal{W}_{e,s}^{B_s}$ and $\overline{A}_s$ on $\overline{D}_s$.)

b) *In IX.2.8, $\overline{B} \cup D$ and $\overline{C} \cup D$ cannot always be taken to be of incomparable Turing degree.* (Jockusch) (Hint: let D be a Turing complete, cointroreducible (II.6.7) but not hyperhypersimple set, e.g. the deficiency set of $\mathcal{K}$. Then, for every set X, if $X \cap \overline{D}$ is infinite, we have $X \cap \overline{D} \equiv_T D$.)

Hyperhypersimple sets

An immediate consequence of the Owings Splitting Theorem is the following result, originally proved by Lachlan by a direct method that inspired IX.2.8.

Theorem IX.2.10 Lattice-Theoretical Characterization of Hyperhypersimplicity (Lachlan [1968b]) *The following are equivalent, for a coinfinite r.e. set A:*

1. *A is hyperhypersimple*
2. *$\mathcal{L}(A)$ is a Boolean algebra*
3. *for every r.e. set $B \supseteq A$, $A \cup \overline{B}$ is r.e.*

Proof. The last two assertions are clearly equivalent, since $B \supseteq A$ has complement in $\mathcal{L}(A)$ if and only if, for some r.e. set $C \supseteq A$, $B \cap C = A$ and $B \cup C = \omega$. Thus $A \cup \overline{B} = C$.

Suppose that, for some $B \supseteq A$, $A \cup \overline{B}$ is not r.e. By IX.2.8 we can split B into r.e. sets B_0 and C such that $A \cup \overline{B}_0$ and $A \cup \overline{C}$ are not r.e. Thus $B_0 \cap \overline{A} \neq \emptyset$. Then we can split C, and so on. By uniformity, we obtain an infinite array of disjoint r.e. sets intersecting $\overline{A}$, and thus A is not hyperhypersimple.

Conversely, suppose A is not hyperhypersimple. By III.4.6, there is a recursive function f such that

- $\mathcal{W}_{f(x)} \cap \mathcal{W}_{f(y)} = \emptyset$ if $x \neq y$
- $\mathcal{W}_{f(x)} \cap \overline{A} \neq \emptyset$.

We want to build an r.e. set $B \supseteq A$ without complement in $\mathcal{L}(A)$, by diagonalization. For each e, we look at $\mathcal{W}_e \cap \overline{A}$ and $\mathcal{W}_{f(e)} \cap \overline{A}$:

- if they are disjoint, by not putting $\mathcal{W}_{f(e)} \cap \overline{A}$ into B we ensure that $\mathcal{W}_e$ does not cover $\overline{B}$
- if they intersect, by putting $\mathcal{W}_e \cap \mathcal{W}_{f(e)}$ into B we ensure that $\mathcal{W}_e$ is not disjoint from B on $\overline{A}$.

It is thus enough to let

$$B = A \cup \bigcup_{e \in \omega} (\mathcal{W}_e \cap \mathcal{W}_{f(e)}). \quad \Box$$

Exercises IX.2.11 a) *There is no coinfinite r.e. set A such that $\mathcal{L}(A)$ is effectively complemented, i.e. such that, for some recursive function f,*

$$\mathcal{W}_x \supseteq A \; \Rightarrow \; \mathcal{W}_{f(x)} \cap \mathcal{W}_x = A \; \wedge \; \mathcal{W}_{f(x)} \cup \mathcal{W}_x = \omega.$$

(Arslanov [1969]) (Hint: let $\mathcal{W}_{h(x)} = \mathcal{W}_x \cup A$. Then fh is a recursive function, and hence it has a fixed-point, contradiction.)

b) *There is no coinfinite r.e. set A such that, for some function $f \leq_T \mathcal{K}$,*

$$\mathcal{W}_x \supseteq A \; \Rightarrow \; \mathcal{W}_{f(x)} \cap \mathcal{W}_x =^* A \; \wedge \; \mathcal{W}_{f(x)} \cup \mathcal{W}_x =^* \omega.$$

(Arslanov, Nadyrov and Soloviev [1977]) (Hint: use the Limit Lemma IV.1.17 to show that if $f \leq_T \mathcal{K}$, then f has an almost fixed-point, i.e. there is e such that $\mathcal{W}_{f(e)} =^* \mathcal{W}_e$.)

Lachlan [1968b] has proved that *the possible isomorphism types of $\mathcal{L}^*(A)$ for A hyperhypersimple are exactly the Σ^0_3 Boolean algebras*, i.e. the (obvious quotients of) Boolean algebras $\langle B, \sqsubseteq, \sqcup, \sqcap, 0, 1\rangle$ such that, for some (not necessarily effective) enumeration $\{b_x\}_{x\in\omega}$ of B, the relation $b_x \sqsubseteq b_y$ is Σ^0_3, and for some recursive functions f and g,

$$b_x \sqcup b_y = b_{f(x,y)} \quad \text{and} \quad b_x \sqcap b_y = b_{g(x,y)}.$$

It is easy to see that these conditions are necessary, since $\mathcal{W}_x \subseteq^* \mathcal{W}_y$ is Σ^0_3 in x and y.

This result completely describes the possible structure of $\mathcal{L}^*(A)$ for A hyperhypersimple (see IX.3.10 for the case of A not hyperhypersimple), and it is proved by an extension of the method used in IX.2.28.f (by representing Boolean algebras in a tree form).

Since the lattice of all hyperhypersimple sets modulo finite sets is relatively complemented (by the same proof as IX.2.10), it follows from Ershov [1964] that *the theory of hyperhypersimple sets is decidable.*

R-maximal sets

The following notion is opposite to that of a hyperhypersimple set.

Definition IX.2.12 *An r.e. set A is* **r-maximal** *if $\mathcal{L}^*(A)$ has no nontrivial complemented elements.*

Exercises IX.2.13 a) *A set A is r-maximal if and only if it is r.e. and has r-cohesive complement* (IX.1.10), i.e. $\overline{A}$ is infinite and cannot be split into two infinite parts by a recursive set. This accounts for the 'r' in the name r-maximal. (Hint: use the Reduction Property II.1.23.)

b) *A set A is r-maximal if and only if it is r.e. and coinfinite, and there is no pair of r.e. sets whose union is ω and whose intersection is A.* (Alton [1975])

c) *A set A is r-maximal if and only if it is simple, and every pair of disjoint, nonrecursive r.e. sets with union A is recursively inseparable.* (Cleave [1970]) (Hint: given A as stated and B recursive, consider $A \cap B$ and $A \cap \overline{B}$. They are recursively separable and r.e., so cannot both be nonrecursive. If e.g. $A \cap B$ is recursive, then $B \cap \overline{A}$ is r.e. and, by simplicity, it is finite.)

d) *Simplicity cannot be dropped in part c).* (Hint: given A r-maximal, consider the set $\{2x : x \in A\}$.)

e) *Maximal $\Rightarrow$ r-maximal $\Rightarrow$ strongly hypersimple.* (Hint: suppose $\{\mathcal{W}_{f(x)}\}_{x\in\omega}$ is a disjoint weak array covering $\overline{A}$. Then $A \cup \bigcup_{x\in\omega} \mathcal{W}_{f(2x)}$ and $A \cup \bigcup_{x\in\omega} \mathcal{W}_{f(2x+1)}$ have union ω and intersection A. Hence A is not r-maximal.)

The existence of r-maximal sets is implied by that of maximal sets (III.4.18). To build different kinds of r-maximal sets, the following notion is useful.

Definition IX.2.14 (Lachlan [1968b]) *An r.e. set B is a* **major subset** *of an r.e. set A if*

1. *$B \subseteq A$*
2. *$A - B$ is infinite*
3. *there is no r.e. way of effectively covering $\overline{A}$ without also effectively covering almost all of $\overline{B}$, i.e. for every e*

$$\overline{A} \subseteq \mathcal{W}_e \ \Rightarrow\ \overline{B} \subseteq^* \mathcal{W}_e.$$

Exercise IX.2.15 *For r.e. sets $B \subseteq A$ differing infinitely, B is a major subset of A if and only if there is no recursive set $R \subseteq A$ such that $R \cap (A - B)$ is infinite.* (Hint: if B is not a major subset of A, use the Reduction Property II.1.23. Conversely, if R exists then either $\overline{R} \cap A$ is finite, and we can consider $\overline{R}$, or $\overline{R} \cap A$ is infinite, and we can consider the complement of an infinite recursive subset of it.)

Natural examples of major subsets of many sets, among which $\mathcal{K}$, will be exhibited in the proof of IX.2.34. We give here a more general direct construction.

Proposition IX.2.16 (Lachlan [1968b]) *Every nonrecursive r.e. set has a major subset.*

Proof. Given A r.e. and nonrecursive, we construct a major subset B of A as in the construction of a maximal set (III.4.18), with two major differences.

First, we do not try to have $\overline{B} \subseteq^* \mathcal{W}_e$ for every infinite r.e. set $\mathcal{W}_e$, but rather only for those such that $\overline{A} \subseteq \mathcal{W}_e$. Thus, instead of using an enumeration of all the r.e. sets we use:

$$U_e = \begin{cases} \mathcal{W}_e & \text{if } \overline{A} \subseteq \mathcal{W}_e \\ \text{finite} & \text{otherwise.} \end{cases}$$

To obtain U_e, it is enough to define

$$x \in U_{e,s} \ \Leftrightarrow\ x \in \mathcal{W}_{e,s} \ \wedge\ (\forall y \leq x)(y \in \mathcal{W}_{e,s} \cup A_s).$$

We will construct $A - B = \{a_0 < a_1 < \cdots\}$ as in III.4.18, and at each stage we will have approximations $A_s - B_s = \{a_0^s < a_1^s < \cdots\}$. The requirements are:

$$\begin{aligned} P_e &\ :\ U_e \text{ infinite } \Rightarrow\ A - B \subseteq^* U_e \\ N_e &\ :\ A - B \text{ has at least } e \text{ elements, i.e. } \lim_{s\to\infty} a_e^s < \infty. \end{aligned}$$

The second difference in the construction of a maximal set is that the elements a_e^s are supposed to enumerate $A - B$, and thus they have to be in A. Since at any stage we only know finite approximations to A, we will only have finitely many of the a_e^s available.

Except for these differences, the construction is as in III.4.18. Precisely, we define the e-state of x at stage s as

$$\sigma(e, x, s) = \sum\{2^{e-i} : i \leq e \ \wedge\ x \in U_{i,s}\}.$$

We start with $B_0 = \emptyset$ and all the a_e^0 undefined. At stage $s+1$ we have A_s, B_s and $A_s - B_s = \{a_0^s < a_1^s < \cdots\}$. By induction, let a_e^{s+1} be the smallest $x \in A_s - B_s$ greater than a_{e-1}^{s+1} and with maximal e-state (if such an element exists).

The construction is effective, and thus B is an r.e. subset of A. Moreover:

- *$A - B$ is infinite*
  Suppose, by induction, that $\lim_{s\to\infty} a_i^s$ exists, for all $i < e$. Let s_0 be a stage after which all these a_i's have reached their final position. Then, for

$s > s_0$, a_e^s may move only to a smaller element that is newly generated in A (and in this case, by construction, the e-state does not decrease), or to reach a higher e-state, and hence only finitely many times.

- *B is a major subset of A*
  We first see that P_0 is satisfied. Suppose $\overline{A} \subseteq \mathcal{W}_0$: then $U_0 = \mathcal{W}_0$ is infinite and, since A is not recursive, $\mathcal{W}_0 \searrow A$ is infinite. Then by construction all elements a_e come sooner or later in $\mathcal{W}_0$, and hence $A - B \subseteq \mathcal{W}_0$.

  In general, the set

  $$\{x : x \in A \ \wedge \ (\forall i \leq e)(U_i \text{ infinite } \Rightarrow x \in U_i \setminus A)\}$$

  is infinite, and thus P_e is satisfied. $\square$

The trick of using special enumerations of the r.e. sets (such as $\{U_e\}_{e\in\omega}$) is very useful in special constructions where only some r.e. sets have to be considered. This is a common tool in building automorphisms of $\boldsymbol{\mathcal{E}}^*$.

Corollary IX.2.17 (Robinson [1967a], Lachlan [1968b]) *There exists an r-maximal set which is not maximal.*

Proof. Let A be a maximal set, and B a major subset of A. Clearly B is not maximal, since both $\overline{A}$ and $A \cap \overline{B}$ are infinite. But B is r-maximal, otherwise some recursive set R would split $A - B$ into two infinite parts (since, by maximality, $\overline{A}$ cannot be split). But, by maximality, one of R and $\overline{R}$ joins $\overline{A}$ to ω, with at most finitely many exceptions. Then B would not be a major subset of A. $\square$

Corollary IX.2.18 Robinson [1968], Lachlan [1968b]) *There exists a hypersimple set which is not r-maximal.*

Proof. Let A be a nonrecursive r.e. set, and B a major subset of A. If f is a recursive one-one function with range A, let $C = f^{-1}(B)$.

$B \cup \overline{A}$ covers $\overline{A}$ but does not cover almost all of $\overline{B}$: since B is a major subset of A, that means that $B \cup \overline{A}$ is not r.e. Then, by the Owings Splitting Theorem, there are two disjoint r.e. sets A_0 and A_1 such that $A = A_0 \cup A_1$, and $B \cup \overline{A}_i$ is not r.e., in particular $\overline{B} \cap A_i$ is infinite. Then $f^{-1}(A_0)$ and $f^{-1}(A_1)$ are complemented in $\mathcal{L}(C)$, and C is not r-maximal.

Moreover, C is hypersimple. Otherwise, let $\{D_{g(x)}\}_{x\in\omega}$ be a strong array intersecting $\overline{C}$; from it we could build a recursive subset of A which splits $A-B$. Its complement would then be an r.e. set covering $\overline{A}$ but not covering almost all of $\overline{B}$. $\square$

IX.2.18 is also a simple consequence of results about degrees, since by III.3.13 every r.e. T-degree contains a hypersimple set, but by IX.2.25.c only high r.e. T-degrees contain r-maximal sets.

Exercises IX.2.19 (Lachlan [1968b]) a) *Any major subset of an r-maximal set is r-maximal.*

b) *If an r-maximal set B is contained in an r-maximal set A and differs infinitely from it, then B is a major subset of A.*

c) *If an r-maximal set is a major subset of a hyperhypersimple set, then the latter is unique (up to finite differences) and maximal.*

Let B be a major subset of A. As already noted in IX.2.18, by the Owings Splitting Theorem it is possible to split A into two r.e. subsets which nontrivially split $A - B$. Thus *the following lattice of r.e. sets restricted to $A - B$*

$$\boldsymbol{\mathcal{E}}^*(A - B) = \{\mathcal{W}_e \cap (A - B)\}_{e \in \omega}$$

is nontrivial (it is actually dense, by IX.2.21.b and IX.2.26). Maass and Stob [1983] have shown that *this lattice is independent of A and B*, and is embedded in every principal filter generated by a nonhyperhypersimple set. For more information on major subsets see Lerman [1971a], Lerman, Shore and Soare [1978], Maass [1985a] and Stob [1985].

A full characterization of *the possible isomorphisms types of $\mathcal{L}^*(A)$ for A r-maximal* is not known. However, there are the following three cases:

1. *A is maximal*
   Then $\mathcal{L}^*(A)$ is uniquely determined, being isomorphic to the two-element Boolean algebra.

2. *A has a proper maximal superset M* (see IX.2.28.e)
   Then any coinfinite r.e. superset of A is almost contained in M, and $\mathcal{L}^*(A)$ is isomorphic to $\boldsymbol{\mathcal{E}}^*(M - A)$ plus a greatest element (corresponding to ω). Moreover, by IX.2.19.b, A is a major subset of M. By the result of Maass and Stob quoted above, $\boldsymbol{\mathcal{E}}^*(M - A)$ is uniquely determined, and hence so is $\mathcal{L}^*(A)$.

3. *A has no maximal superset* (see IX.2.28.d)
   Robinson [1967a] and Lachlan [1968b] have produced two examples with distinct isomorphism types, and Cholak and Nies [199?] have proved that there are infinitely many such types.

Maximal sets

By definition, a coinfinite r.e. set is maximal if and only if it has only trivial r.e. supersets, i.e. those either differing finitely from it or cofinite. In lattice-theoretical terms this is equivalent to saying that $\mathcal{L}^*(A)$ is isomorphic to the trivial Boolean algebra $\{0,1\}$, or that the equivalence class of A is a coatom (i.e. a maximal element) in $\boldsymbol{\mathcal{E}^*}$, and the latter characterization justifies the name 'maximal'.

The next observation relates in a nontrivial way the three notions dealt with so far.

Proposition IX.2.20 (Lachlan [1968b]) *An r.e. set is maximal if and only if it is both hyperhypersimple and r-maximal.*

Proof. A maximal set is obviously both hyperhypersimple and r-maximal. Conversely, if A is hyperhypersimple and r-maximal then $\mathcal{L}^*(A)$ is a Boolean algebra by IX.2.10, and without nontrivial complements by IX.2.12. Then $\mathcal{L}^*(A)$ is isomorphic to $\{0,1\}$, and A is maximal. $\Box$

IX.2.17 provides an example of an r-maximal set which is not maximal, and hence also not hyperhypersimple. A hyperhypersimple set which is not maximal, and hence also not r-maximal, was exhibited in III.4.20.2.

Exercises IX.2.21 Maximal subsets. (Lachlan [1968b]) An r.e. set B is a **maximal subset** of an r.e. set A if $B \subseteq A$, and $A - B$ is cohesive, i.e. it is infinite and there is no r.e. set which splits it nontrivially.

a) *Every infinite r.e. set has a maximal subset.* (Hint: let f be a recursive one-one function enumerating the given set, and consider $f(A)$ with A maximal.)

b) *No subset of an r.e. set can be both maximal and major.* (Hint: see the proof of IX.2.18.)

c) *Every maximal subset of a maximal set A is the intersection of A with a maximal set.* (Hint: let B be a maximal subset of A. If $B \cup \overline{A}$ is not r.e. then, by the Owings Splitting Theorem, B is not a maximal subset of A. So $B \cup \overline{A}$ is r.e. and then maximal, and $B = A \cap (B \cup \overline{A})$.)

Exercises IX.2.22 Quasimaximal sets. A set A is **quasimaximal** if it is the intersection of finitely many maximal sets.

a) *A coinfinite r.e. set A is quasimaximal if and only if $\mathcal{L}^*(A)$ is finite.*

b) *maximal $\Rightarrow$ quasimaximal $\Rightarrow$ hyperhypersimple.*

c) *There are quasimaximal, nonmaximal sets.* (Hint: intersect two maximal sets differing infinitely. See also III.4.20.)

d) *There are hyperhypersimple, nonquasimaximal sets.* (Hint: a hyperhypersimple set not contained in a maximal set, which exists by IX.2.28.f, is such a set.)

We now turn to a nice result that characterizes the class of r.e. degrees containing maximal sets, in terms of a classification of degrees introduced in XI.1.9. But first we prove a variation of XI.1.10 that will simplify the proof.

Proposition IX.2.23 (Robinson [1968]) *If $\boldsymbol{a}$ is a high r.e. degree, then there is an r.e. set $B \in \boldsymbol{a}$ and a recursive enumeration $\{B_s\}_{s\in\omega}$ of it such that the following (version of the computation) function*

$$C_B(x) = \mu s.\,(\forall y \leq x)(y \in B_s \ \Leftrightarrow\ y \in B)$$

dominates every total recursive function.

Proof. Let A be an r.e. set in the given degree $\boldsymbol{a}$. Since $\boldsymbol{a}$ is high, by XI.1.10 there is a function $f \leq_T A$ which dominates every total recursive function. Let $f \simeq \varphi_e^A$, and

$$g(x,s) = \begin{cases} \varphi_{e,s}^{A_s}(x) & \text{if } \varphi_{e,s}^{A_s}(x)\downarrow \\ 0 & \text{otherwise.} \end{cases}$$

Then the modulus of g

$$m_g(x) = \mu s.\,(\forall t \geq s)[g(x,t) = g(x,s)]$$

is recursive in A, and it still dominates every total recursive function because f does (notice that if $m_g(x) \leq h(x)$ then $f(x) \leq g(x, h(x))$).

We now build $\overline{B} = \{b_0 < b_1 < \cdots\}$ as required. To have $A \leq_T B$, we code A into B. To have $B \leq_T A$, we use permitting (III.3.16). And to have $m_g \leq_T C_B$, we leave a trace in B whenever g changes.

The construction is the following. We start with $B_0 = \emptyset$. At stage $s+1$ we have $\overline{B}_s = \{b_0^s < b_1^s < \cdots\}$, and put b_x^s in B if one of the following happens:

- $x \in A_{s+1} - A_s$
- $x \leq s$ and $g(x, s+1) \neq g(x,s)$.

Since f is total, and thus g only changes finitely often on x, $\lim_{s\to\infty} b_x^s$ exists, and thus C_B is total as well. Moreover, C_B dominates every recursive function because m_g does, and we made sure that the approximation of B changes at least as many times as g does.

$A \leq_T B$ because to determine whether $x \in A$, it is enough to look, recursively in B, for a stage s such that $b_x = b_x^s$. Then, by construction, $x \in A$ if and only if $x \in A_s$.

$B \leq_T A$ because to determine a stage s such that $b_x = b_x^s$, it is enough to look, recursively in A, for a stage s greater than $m_g(x)$ (so that g cannot change after it) and such that A has settled on all $y \leq x$. $\square$

Theorem IX.2.24 (Martin [1966a]) *An r.e. degree contains a maximal set if and only if it is high.*

Proof. If A is maximal, then it is dense simple (III.4.15), and hence the function

$$f(x) = \text{the } x\text{-th element of } \overline{A} \text{ in the order by magnitude}$$

dominates every total recursive function. Since $f \leq_T A$, from XI.1.10 it follows that the degree of A is high.

Given an r.e. high degree, by IX.2.23 there is an r.e. set B in it such that the function

$$C_B(x) = \mu s.\,(\forall y \leq x)(y \in B_s \;\Leftrightarrow\; y \in B)$$

dominates every total recursive function. We now build a maximal set A such that $A \equiv_T B$. The construction is a modification of that for a maximal set (III.4.18), in particular we build the complement of A, and at stage s we have $\overline{A}_s = \{a_0^s < a_1^s < \cdots\}$.

To obtain $A \leq_T B$, we use permitting (III.3.16), i.e. we put elements in A at any given stage only if some smaller element of B is generated at the same stage. Thus we will not always be able to put elements into A for the sake of maximizing e-states.

To obtain $B \leq_T A$, we leave a trace in A whenever $e \in B_{s+1} - B_s$, e.g. we put a_e^s (the e-th element of $\overline{A}_s$) into A. To see if $e \in B$ recursively in A, we search for s such that the first $s+1$ elements of $\overline{A}$ have settled down, and thus a_e^s is final. Then $e \in B$ if and only if $e \in B_{s+1}$, otherwise a_e^s would enter A after stage $s+1$, contradicting the fact that it reached its final position. For a technical reason that will be clear in the following, the coding strategy is more complicated: whenever $e \in B_{s+1} - B_s$ we put into A whichever of a_e^s and a_{e+1}^s has a lower e-state. Note that in any case this does not interfere with permitting, since $a_{e+1}^s > a_e^s \geq e$ for any stage s.

The construction is as follows. We start with $A_0 = \emptyset$. We alternately take care of maximality requirements and of the coding strategy. We may suppose that B generates elements only at even stages. At stage $s+1$:

- *s even*
  Let e_0 be the greatest e such that a_e^s is not permitted to enter A, i.e. it is not greater than the unique element in $B_{s+2} - B_s$. Then we let $a_e^{s+1} = a_e^s$ if $e \leq e_0$, since these elements cannot go into A. For $e > e_0$, we follow the usual strategy, i.e. we let a_e^{s+1} be the smallest $x \in \overline{A}_s$ greater than a_{e-1}^{s+1} and with maximal e-state.

- *s odd*
  Let $e_0 \in B_{s+1} - B_s$ be the element generated by B at stage $s+1$ (which is even). Then $a_e^{s+1} = a_e^s$ if $e < e_0$, $a_{e_0}^{s+1}$ is whichever of $a_{e_0}^s$ and $a_{e_0+1}^s$ has higher e-state (if they have the same e-state, either of them will do), and $a_e^{s+1} = a_{e+1}^s$ if $e > e_0$.

Now $A \leq_T B$ by permitting, and $B \leq_T A$ by the coding strategy. It remains to be proved that A is maximal.

- *$\overline{A}$ is infinite*
  Note that an element a_e^s moves for the coding only finitely many times, and otherwise it moves only to reach higher e-states. Thus it moves only finitely often.

- *A is maximal*
  We show that the requirements

$$P_e \quad : \quad \mathcal{W}_e \cap \overline{A} \text{ finite or } \overline{\mathcal{W}}_e \cap \overline{A} \text{ finite}$$

  are satisfied, by induction on e. Suppose it is so for $i < e$, i.e. each $\mathcal{W}_i$ is almost constant on $\overline{A}$. We show that if $\mathcal{W}_e \cap \overline{A}$ and $\overline{\mathcal{W}}_e \cap \overline{A}$ are both infinite, we would have a recursive function h not dominated by C_B.

  Consider the function $h(x)$ that gives the smallest s such that one of the following holds:

  - $x \in A_s$
  - $x \in \mathcal{W}_{e,s}$
  - $x \in \overline{A}_s$, the $(e-1)$-state of x at stage s is the final one, and there is $y > x$ in $\overline{A}_s$ with higher e-state.

  h is defined for almost every x, precisely for every x larger than the smallest element of $\overline{A}$ such that after it, all elements have the final $(e-1)$-state.

  Note that, given $x \in \overline{\mathcal{W}}_e \cap \overline{A}$, for every $t \geq h(x)$ there are elements of $\overline{A}_t$ greater than x with higher e-state at t. Indeed, a $y > x$ can enter A only because either there are elements with greater e-state to which we move (but this does not make the e-state decrease), or for the coding. Since we used two elements for the coding, if there was a single element greater than x with higher e-state we avoided putting it into A, and thus the claim holds. This is where the modified coding is crucial.

  Suppose $x = a_i$, for some $i \geq e$. If x were permitted after stage $h(x)$, it could go into A. Since it does not, it must be the case that B settles on every element up to x before stage $h(x)$, i.e. $C_B(x) \leq h(x)$. This holds for almost every $x \in \overline{\mathcal{W}}_e \cap \overline{A}$, in particular for infinitely many elements. Then C_B does not dominate the recursive function h, which is a contradiction.
  $\square$

Exercises IX.2.25 a) *The construction automatically gives $A \equiv_T B$, even without the coding step.* (Jockusch) (Hint: use the Fixed-Point Theorem as in III.3.19.b.)

b) *An r.e. degree contains a hyperhypersimple set if and only if it is high.* (Martin [1966a]) (Hint: it is enough to show, as first proved by Martin [1963], that *a hyperhypersimple set is dense simple*. Let A be hyperhypersimple, $\overline{A} = \{a_0 < a_1 < \cdots\}$, and suppose A is not dense simple, i.e. there is a recursive function f such that, for infinitely many n, $a_n \leq f(n)$. Since A is hypersimple, no recursive function dominates $\overline{A}$, i.e. $a_n \leq f(n)$ cannot hold for almost every n. Hence there are infinitely many n such that

$$f(n) < a_n < a_{n+1} \leq f(n+1).$$

We contradict the fact that A is hyperhypersimple by building a disjoint weak array $\{B_i\}_{i\in\omega}$ intersecting $\overline{A}$. For every n, wait until a stage s_n comes such that only n numbers are in $\overline{A}_s$ and below $f(n)$. At any stage $s \geq s_n$, take the least i such that $B_{i,s} \cap \{0, \ldots, f(n)\} \subseteq A_s$, and try to make $B_i \cap \overline{A} \neq \emptyset$ by putting in it the smallest $y \in \overline{A}_s$ between $f(n)$ and $f(n+1)$ which is not yet in any of the B_j's.)

c) *An r.e. degree contains an r-maximal set if and only if it is high.* (Martin [1966a]) (Hint: this is not immediate, since r-maximal sets are not always dense simple. Let A be not of a high degree. To show that A is not r-maximal, we want to build two disjoint recursive sets R_0 and R_1, each intersecting $\overline{A}$ infinitely often. To make them recursive, it is enough to enumerate them in such a way that at stage s, only elements greater than s go into them. At each stage, put one new element in each of them, so that there are exactly $2s$ elements in $R_{0,s-1} \cup R_{1,s-1}$. Let $f(s)$ be the least t such that the least $2s+2$ elements of $\overline{A}_t$ greater than s are in $\overline{A}$. Since f is recursive in A, and A is not of a high degree, by XI.1.10 there is a recursive function g such that $f(s) \leq g(s)$ for infinitely many s. The construction is thus as follows: at stage s, take the least two elements greater than s and not in $A_{g(s)} \cup R_{0,s-1} \cup R_{1,s-1}$, and put one in R_0 and the other in R_1.)

d) *There exists an incomplete maximal set.* (Sacks [1964b]) (Hint: in the construction of a maximal set, use restraints to ensure $C \not\leq_T A$ for a given nonrecursive r.e. set C. This requires the infinite injury method of Chapter X. In the notation of Section X.3, use the Injury and Window Lemmas to show, by induction on e, that P_e is satisfied, $\hat{I}_{e+1}$ is recursive, and N_{e+1} is satisfied. An indirect proof uses IX.2.24 and the existence of incomplete high r.e. degrees instead, see XI.1.13.a.)

Lerman [1970] and Kobzev [1973a] have proved that *every r.e. degree containing maximal sets actually contains infinitely many tt-degrees of maximal sets*. Yates [1969] and Marchenkov [1976b] have proved that, for any reducibility between $\leq_m$ and $\leq_T$, given an incomplete, nonrecursive r.e. set A there is a maximal set incomparable with it.

Alton [1974] has considered the lattice $\boldsymbol{\mathcal{E}}^\alpha$ obtained from $\boldsymbol{\mathcal{E}}$ by successively factoring α times by the filter generated by maximal elements, and has called **$\boldsymbol{\alpha}$-maximal** the maximal elements of $\boldsymbol{\mathcal{E}}^\alpha$. He shows that *$\alpha$-maximal sets exist if and only if* $\alpha < \omega_1^{ck}$ (where ω_1^{ck} is the first nonrecursive ordinal, see p. I.384).

Sets without maximal supersets

The next observation connects the property of not having maximal supersets to the filter structure.

Proposition IX.2.26 (Myhill [1956], Lachlan) *An r.e. set A has no maximal superset if and only if $\mathcal{L}^*(A)$ is dense.*

Proof. One direction is trivial because the equivalence classes of maximal sets are maximal elements in $\boldsymbol{\mathcal{E}^*}$, and thus if A has a maximal superset then $\mathcal{L}^*(A)$ is not dense (actually, it has coatoms).

Conversely, suppose $\mathcal{L}^*(A)$ is not dense. There are r.e. sets B and C such that $A \subseteq B \subseteq C$, and with B a maximal subset of C. We prove that $B \cup \overline{C}$ is a maximal set including A. The only thing to note is that it is r.e. If not, the proof of IX.2.10 would show that its complement $C - B$ is not hyperhyperimmune, contradicting the fact that it is cohesive. $\square$

In particular, $\mathcal{E}$ is dense if and only if there is no maximal set, although this can easily be proved directly.

We turn now to existence results. We have already noted in III.4.24.d that *Post's simple set has no maximal superset* (Cohen and Jockusch [1975]), and another natural example is shown in IX.2.30. We now give a direct construction.

Proposition IX.2.27 (Martin [1963]) *There exists a coinfinite r.e. set without maximal supersets.*

Proof. Notice that if B is maximal, then there is no recursive function h such that if

$$I_n = \{x : h(n) \leq x < h(n+1)\}$$

then, for infinitely many n, $I_n \cap \overline{B}$ has at least two elements. Otherwise, we could build an r.e. set by putting into it the elements of each I_n up to and including the first not in B, and we could thus split $\overline{B}$ into two infinite parts.

We now build A r.e. and h recursive such that, for every coinfinite r.e. set $B \supseteq A$, there are infinitely many n such that $I_n \cap \overline{B}$ has at least two elements; then B is not maximal.

If B is coinfinite, for infinitely many n we will have $I_n \cap \overline{B} \neq \emptyset$, so it is enough to ensure by construction that $I_n \cap \overline{B}$ does not have exactly one element, with at most finitely many exceptions. Since we are interested in $B \supseteq A$ and thus $\overline{B} \subseteq \overline{A}$, the construction will be such that when we see that $I_n \cap \overline{\mathcal{W}}_e$ has only one element, we put this element into A.

Since we want A coinfinite, we cannot let every $\overline{\mathcal{W}}_e$ interfere with every I_n, and we thus allow it to do so only for $n \geq e$. Since every $\overline{\mathcal{W}}_e$ contributes at most one element of I_n to A, A is coinfinite if I_n has at least $n+2$ elements. We thus let

$$h(0) = 0 \quad \text{and} \quad h(n+1) = h(n) + n + 2.$$

The construction is as follows. We start with $A_0 = \emptyset$. At stage $s+1$, for every e and $n \geq e$ such that $I_n \cap \overline{\mathcal{W}}_{e,s}$ has only one element, we put this element into A.

If $B \supseteq A$ is r.e. and coinfinite, for infinitely many n we have $I_n \cap \overline{B} \neq \emptyset$. If $B = \mathcal{W}_e$, $n \geq e$, and $I_n \cap \overline{B} \neq \emptyset$, then by construction $I_n \cap \overline{B}$ has at least two elements. Thus B is not maximal. $\square$

Exercises IX.2.28 R.e. sets with(out) maximal supersets.

a) *If A is a nonrecursive r.e. set, B is a major subset of A and f is a recursive one-one function with range A, then $f^{-1}(B)$ has no maximal superset.* (Lachlan [1968b]) (Hint: see IX.2.18 and IX.2.21.b.)

b) *Every coinfinite, nonsimple r.e. set is contained in a maximal set.* (Ullian [1961]) (Hint: if $\overline{A}$ is infinite and it contains an infinite recursive set B, let f be a recursive one-one function with range B, and consider the union of $\overline{B}$ and the image of a maximal set via f.)

c) *There are simple, nonhypersimple sets without maximal supersets.* (Martin [1963]) (Hint: see the proof of IX.2.27.)

d) *There exists an r-maximal set without maximal supersets.* (Robinson [1967a], Lachlan [1968b]) (Hint: build an r.e. set A and a sequence $\{T_e\}_{e\in\omega}$ of r.e. sets, not necessarily uniformly r.e., such that $A \subseteq T_0 \subseteq T_1 \subseteq \cdots$, each set differs infinitely from the following one and, for every e, $\mathcal{W}_e \subseteq T_e$ or $\overline{A} \subseteq^* \mathcal{W}_e$. Suppose $\mathcal{W}_e \supseteq A$ is coinfinite; then it cannot be $\overline{A} \subseteq^* \mathcal{W}_e$, hence $\mathcal{W}_e \subseteq T_e \subseteq T_{e+1}$ and $\mathcal{W}_e$ is not maximal. To show that A is r-maximal, suppose that $R = \mathcal{W}_e = \overline{\mathcal{W}_i}$, for some $i > e$, splits $\overline{A}$ into two nontrivial parts. Then it cannot be $\overline{A} \subseteq^* \mathcal{W}_e$; so $\mathcal{W}_e \subseteq T_e$, and $\mathcal{W}_e \cap \overline{A}$ is infinite. But then it cannot be $\mathcal{W}_i \subseteq T_i$ or $\overline{A} \subseteq^* \mathcal{W}_i$. To build A and the T_e's, use a *planar version of the e-state method*.) A different construction is given in the proof of part 1 on p. 393.

e) *There exists a nonmaximal r-maximal set contained in a maximal set.* (Lachlan [1968b]) (Hint: see the proof of IX.2.17.)

f) *There exists a hyperhypersimple set without maximal supersets.* (Lachlan [1968b]) (Hint: consider

$$\overline{A} = \{a_\emptyset < a_0 < a_1 < a_{00} < a_{01} < a_{10} < a_{11} < \cdots\}$$

and $C_\sigma = \{a_\tau : \tau \supseteq \sigma\}$. Thus the complement of A is pictured in a tree form, and C_σ is the part of it determined by the subtree with vertex a_σ. The requirements are:

$$\begin{aligned} P_e &: \; |\sigma| = e \Rightarrow (\forall i \leq e)(\mathcal{W}_i \cap C_\sigma =^* \emptyset \text{ or } C_\sigma \subseteq^* \mathcal{W}_i) \\ N_e &: \; A \cup C_e \text{ is r.e.} \end{aligned}$$

Then A is hyperhypersimple because $\mathcal{L}(A)$ is a Boolean algebra, since the complement of $\mathcal{W}_e$ can be obtained from the vertices of level e. Since P_e is satisfied, $\mathcal{W}_e$ is such that it either contains a cone or is disjoint from it, with at most finitely many exceptions. Consider the $C_{\sigma_1}, \dots, C_{\sigma_n}$ such that $\mathcal{W}_e$ is disjoint from them; they are r.e. (together with A), since the negative requirements are satisfied, and their union (together with A) gives the complement of $\mathcal{W}_e$ in $\mathcal{L}(A)$. Moreover, A has no maximal superset because if $\mathcal{W}_e \supseteq A$ is coinfinite, then there is at least one σ such that C_σ is disjoint from $\mathcal{W}_e$, and we can split $\overline{\mathcal{W}_e}$ by using $C_{\sigma * 0}$ and $C_{\sigma * 1}$. To build A, use a *tree version of the e-state method*, by playing at stage $s+1$ the usual strategy for P_e within each cone C^s_σ such that $|\sigma| = e$.)

g) *There exists a nonmaximal hyperhypersimple set contained in a maximal set.* (Yates [1962]) (Hint: see III.4.20.)

The next result produces a great number of examples of r.e. sets without maximal supersets, in terms of a classification of degrees introduced in XI.1.9.

Theorem IX.2.29 (Martin [1966a], Lachlan [1968d], Shoenfield [1976]) *Any non-low$_2$ r.e. degree contains an r.e. set without maximal (actually, hyperhypersimple) supersets.*

Proof. Let A be an infinite r.e. set, and f be a recursive one-one enumeration of it: from III.4.12 we know that the deficiency set of f defined by

$$x \in B \Leftrightarrow (\exists y > x)(f(y) < f(x)),$$

is in the same degree as A, and is not hyperhypersimple. However, the proof given there of the latter fact was quite indirect: $\overline{B}$ is retraceable by II.6.16, but the complement of a hyperhypersimple set cannot contain a retraceable subset by III.4.7. We now produce a more direct proof showing that B is not hyperhypersimple, and then generalize it to any coinfinite r.e. superset of B, if in addition A is non-low$_2$.

- *B is not hyperhypersimple*
  To show that B is not hyperhypersimple, we build a disjoint weak array intersecting $\overline{B}$. The obvious idea is to consider the function

  $$g(x) = \text{the smallest } z > x \text{ such that } z \in \overline{B},$$

  which is total because $\overline{B}$ is infinite, and the sets

  $$F_0 = \{0, \dots, g(0)\}, \qquad F_1 = \{g(0)+1, \dots, g(g(0))\}, \qquad \cdots$$

  The first problem is that g is only recursive in B, instead of being recursive, and the F_n's are only recursive in B, instead of being r.e.: the solution is to approximate g, by taking an i such that $g \simeq \{i\}^B$, and considering the function

  $$h(x, s) = \{i\}^{B_s}_s(x),$$

and the sets

$$F_{0,s} = \{0, \ldots, h(0,s)\}, \qquad F_{1,s} = \{h(0,s)+1, \ldots, h(h(0,s),s)\}, \qquad \cdots$$

The second problem is now that, since the $F_{n,s}$'s are not increasing in s, their limits $\{F_n\}_{n\in\omega}$ are Δ^0_2, but still not r.e.: the solution is to use them only as reference, and to define an array $\{G_n\}_{n\in\omega}$ as follows. At stage $s+1$, for each $n \leq s$, if $G_{n,s} \cap \overline{B}_s = \emptyset$ (i.e. if the current approximation of G_n does not intersect the current approximation of $\overline{B}$), we put into $G_{n,s+1}$ the smallest element (if it exists) which is less than s (for effectiveness of the construction), in $F_{n,s} \cap \overline{B}_s$, and not already in any $G_{m,s}$ for $m \neq n$, nor in any $G_{m,s+1}$ for $m < n$ (i.e. not in any other member so far defined, to get a disjoint array).

The third problem is that there is no reason to think that the strategy will succeed: by the time we reach a stage s such that $F_{n,s}$ is final, it could be that the elements of $\overline{B}$ that are in it (and there is at least one by the choice of g) have already been used in the construction, so that we cannot make $G_{n,s} \cap \overline{B} \neq \emptyset$, unless it is already so. The solution is two-fold: on the one hand, to give more chances to G_n, by letting it pick its elements not only from F_n, but from any F_e with $e \geq n$; on the other hand, to put at least $e+1$ elements of $\overline{B}$ into F_e, so that all the G_n's with $n \leq e$ can get one.

We thus redefine g as

$$g(x) = \text{the smallest } z > x \text{ such that } |[x+1, z] \cap \overline{B}| \geq x+1,$$

and approximate it by a recursive function h as above. At stage $s+1$, for each $e \leq s$ and $n \leq e$, if $G_{n,s} \cap \overline{B}_s = \emptyset$ we put into $G_{n,s+1}$ the smallest element (if it exists) which is less than s, in $F_{e,s} \cap \overline{B}_s$, and not already in any $G_{m,s}$ for $m \neq n$, nor in any $G_{m,s+1}$ for $m < n$.

We now prove that the construction works, i.e. that $G_n \cap \overline{B} \neq \emptyset$. For convenience of notation, we let

$$\gamma_{0,s} = 0, \qquad \gamma_{e+1,s} = h(\gamma_{e,s}, s) \qquad \text{and} \qquad \gamma_e = \lim_{s\to\infty} \gamma_{e,s},$$

so that, for $e > 0$,

$$F_{e,s} = \{\gamma_{e,s}+1, \ldots, \gamma_{e+1,s}\} \qquad \text{and} \qquad F_e = \{\gamma_e + 1, \ldots, \gamma_{e+1}\}.$$

We can suppose that $\gamma_{e,s}$ is monotone in s, and that if s_e is the first stage s such that $\gamma_{e,s} = \gamma_e$, then $\gamma_e = s_e$, e.g. by redefining h as

$$h(x, s+1) = \begin{cases} s+1 & \text{if } \{i\}^{B_{s+1}}_{s+1}(x)\downarrow \not\simeq \{i\}^{B_s}_s(x)\downarrow \\ h(x,s) & \text{otherwise.} \end{cases}$$

This is possible because the only property of h we need is that it is large enough, and by convention any element of a computation that converges at stage $s+1$, including the output, is smaller than $s+1$.

By construction, no element greater than s can get into any $G_{n,s}$: thus no element of

$$F_{e,s_e} = \{\gamma_e + 1, \dots, \gamma_{e+1,s_e}\}$$

has been used yet at stage $s_e = \gamma_e$, and it is enough to show that F_{e,s_e} contains at least $e+1$ elements of $\overline{B}$ (since then the construction ensures that $G_n \cap \overline{B} \neq \emptyset$, for all $n \leq e$). By the choice of g and h, and since $e \leq \gamma_e$, this is true of

$$F_e = \{\gamma_e + 1, \dots, \gamma_{e+1}\}.$$

Thus it is enough to prove that if $\gamma_{e+1,s_e} \neq \gamma_{e+1}$, then

$$\{\gamma_{e+1,s_e} + 1, \dots, \gamma_{e+1}\} \subseteq B.$$

Here we finally use the fact that B is a deficiency set, through the following property: if we define its approximations as

$$x \in B_s \Leftrightarrow s > x \wedge f(s) < f(x),$$

then if an element x that is not yet in B enters it at stage s, so does every element z between x and s. Indeed, if $x < z < s$ and x has not entered B before stage s, in particular it has not entered B at stage z, i.e. $f(x) \leq f(z)$; but if x enters B at stage s, then $f(s) < f(x)$; thus $f(s) < f(x) \leq f(z)$, i.e. z enters B at stage s, too.

If $\gamma_{e+1,s_e} \neq \gamma_{e+1}$, then $\{i\}_{s_e}^{B_{s_e}}(\gamma_{e,s_e})$ has changed after stage s_e, say at stage t. Since γ_{e,s_e} has reached its final value at stage s_e, the only possibility for the computation to have changed is that some element below s_e has entered B at stage t: by the property of B just discussed, then every element between s_e and t has also entered B. Since the same holds any time the approximation of γ_{e+1} changes, this proves the claim that $\{\gamma_{e+1,s_e} + 1, \dots, \gamma_{e+1}\} \subseteq B$, and concludes the proof that B is not hyperhypersimple.

- *if A is non-low$_2$, no coinfinite r.e. superset of B is hyperhypersimple*
  Suppose now C is a coinfinite r.e. superset of B: to prove that C is not hyperhypersimple, we build a disjoint weak array intersecting $\overline{C}$ as above, by considering

$$g(x) = \text{the smallest } z > x \text{ such that } |[x+1, z] \cap \overline{C}| \geq x+1.$$

Now g is recursive in C, but it is not of much help to define h as a recursive approximation of it, since the set C will not have in general the property of B that made the previous proof work.

But since g is recursive in $\mathcal{K}$, and (A, and hence) B is non-low$_2$, by XI.1.3 there is a function recursive in B that is not dominated by g, and which is then infinitely often greater than g: we let h be a recursive approximation of *this* function, and then proceed as in the previous proof (using C in place of B).

We can no longer prove that, for *all* e, $G_n \cap \overline{C} \neq \emptyset$ for all $n \leq e$, but we can still prove it for *infinitely many* e: and this is still enough to deduce that $G_n \cap \overline{C} \neq \emptyset$ for all n, as needed.

To prove the claim, we cannot argue directly that, simply because h approximates a function that is greater than g infinitely often, then there are infinitely many e such that F_e contains $e+1$ elements of $\overline{C}$: indeed, the arguments on which h is greater than g are not necessarily among the γ_e's.

Consider thus any x such that, for any sufficiently big s, $g(x) < h(x,s)$: there is an e such that $x \in F_e$, i.e.

$$\gamma_e < x \leq \gamma_{e+1}.$$

We can suppose h monotone in x, by possibly maximizing its values: then, by the last inequality and the definition of γ_{e+2}, for any sufficiently big s we have

$$h(x,s) \leq h(\gamma_{e+1}, s) = \gamma_{e+2}.$$

Putting everything together we get

$$\gamma_e < x < g(x) < \gamma_{e+2}.$$

Between x and $g(x)$ there are at least $x+1$ (and, since $e < x$, at least $e+2$) elements of $\overline{C}$: since we don't know where γ_{e+1} is, in the worst case only half of the elements of $\overline{C}$ are between γ_e and γ_{e+1}, i.e. in F_e, and the other half are between γ_{e+1} and γ_{e+2}, i.e. in F_{e+1}. So we are off by half of the elements we need: but this is easily fixed if we modify the original function g from which we started, and double the number of elements of $\overline{C}$ it picks out, by letting

$$g(x) = \text{the smallest } z > x \text{ such that } |[x+1, z] \cap \overline{C}| \geq 2x+2.$$

With this new function the previous argument shows that, for each x as above, at least one of F_e and F_{e+1} contains the right number of elements of $\overline{C}$, and the proof of the first part goes through. $\Box$

Shoenfield originally proved IX.2.29 by a quite mysterious proof, showing that if B has a hyperhypersimple superset then every $\Pi_2^{0,A}$ set is Δ_3^0, and hence A is low$_2$. The direct proof given above is from Downey and Shore [1995].

Corollary IX.2.30 (Robinson [1968]) *The deficiency set of $\mathcal{K}$ has no maximal (and, actually, no hyperhypersimple) superset.*

Robinson [1966] has proved that every coinfinite low r.e. set has a maximal superset, and Lachlan [1968d] has extended this to every coinfinite low$_2$ r.e. set. This latter result, together with the one just proved, shows that *an r.e. degree contains a coinfinite r.e. set without maximal (or hyperhypersimple) supersets if and only if it is not low$_2$.*

Exercise IX.2.31 *An η-hyperhypersimple semirecursive set is low$_2$.* (Miller [1981]) (Hint: in the proof of IX.2.29, the fact that B is a deficiency set is used only to derive a typical consequence of semirecursiveness.)

We note on p. 400 that the deficiency set of $\mathcal{K}$ is a natural example of an r.e. set without r-maximal supersets. Here we give here a direct construction, in the style of IX.2.27.

Proposition IX.2.32 (Robinson [1967a], Lachlan [1968b]) *There exists a coinfinite r.e. set without r-maximal supersets.*

Proof. Notice that if B is r-maximal then there is no recursive function h such that if

$$I_n = \{x : h(n) \leq x < h(n+1)\}$$

then, for some fixed $a \geq 2$:

1. $|I_n \cap \overline{B}| \leq a$ for almost every n
2. $|I_n \cap \overline{B}| = a$ for infinitely many n

(Downey and Shore [1995]). Otherwise, we could build a recursive set R as follows: to determine $I_n \cap R$, generate B up to the first stage s such that $|I_n \cap \overline{B}_s| \leq a$, put the smallest element of $I_n \cap \overline{B}_s$ into R, and the rest of I_n into $\overline{R}$.

R is recursive because the construction is defined for almost every n, by condition 1 (and R can be defined arbitrarily on the finitely many I_n's for which condition 1 is not satisfied).

R splits $\overline{B}$ non trivially, because by condition 2 there are infinitely many n such that $|I_n \cap \overline{B}| = a$, and thus at the first stage s such that $|I_n \cap \overline{B}_s| = a$ every element of $I_n \cap \overline{B}_s$ is really in $\overline{B}$. And the construction puts one in R, and at least one in $\overline{R}$ (because $a \geq 2$).

We now build A r.e. and, for each e, h_e recursive such that if

$$I_n^e = \{x : h_e(n) \leq x < h_e(n+1)\}$$

and $\mathcal{W}_e \supseteq A$ is coinfinite then, for some fixed $a \geq 2$:

1. $|I_n^e \cap \overline{\mathcal{W}}_e| \leq a$ for almost every n
2. $|I_n \cap \overline{\mathcal{W}}_e| = a$ for infinitely many n.

Thus $\mathcal{W}_e$ is not r-maximal.

Condition 2 is achieved exactly as in IX.2.27: one avoids the troublesome case $|I_n^e \cap \overline{\mathcal{W}}_e| = 1$, by acting when one sees only one element in $|I_n \cap \overline{\mathcal{W}}_e|$, and putting that element into A.

Condition 1 is achieved by putting a fixed bound on $|I_n^e|$, independent of e. For a single r.e. set B, e.g. $\mathcal{W}_0$, a bound 2 would be enough, since the only requirement on a is that $a \geq 2$. Thus we can let

$$h_0(n) = 2n.$$

For $\mathcal{W}_1$ we also need at least two free elements in each interval I_n^1, but now one element of each I_n^0 might go into A because of action for $\mathcal{W}_0$. So we can let

$$h_1(n) = 4n,$$

and in general

$$h_e(n) = 2^{e+1}n.$$

In other words, the intervals I_n^e have 2^{e+1} elements each.

The construction is the following. We start with $A_0 = \emptyset$. At stage $s+1$, for every e and $n \geq e$ such that $I_n^e \cap \overline{\mathcal{W}}_{e,s}$ has only one element, we put this element into A.

$\overline{A}$ is coinfinite because the number of elements of I_e^e that can go into A are:

- 1 for the sake of $\mathcal{W}_e$
- 2 for the sake of $\mathcal{W}_{e-1}$ (since each e-interval consists of 2 $(e-1)$-intervals)
- $\cdots$
- 2^e for the sake of $\mathcal{W}_0$ (since each e-interval consists of 2^e 0-intervals)

and

$$1 + 2 + \cdots + 2^e = 2^{e+1} - 1,$$

while I_e^e has 2^{e+1} elements by definition. Thus the intervals I_e^e, which are all disjoint, contain at least one element of $\overline{A}$ each.

If $\mathcal{W}_e \supseteq A$ is r.e. and coinfinite, for infinitely many n we have $I_n^e \cap \overline{\mathcal{W}}_e \neq \emptyset$, and hence $|I_n^e \cap \overline{\mathcal{W}}_e| \geq 2$. Let a be the greatest number $\leq 2^{e+1}$ such that $|I_n^e \cap \overline{\mathcal{W}}_e| = a$ infinitely often. Then condition 1 is satisfied because a is the greatest such, and condition 2 is satisfied by definition. Then $\mathcal{W}_e$ is not r-maximal. $\Box$

The class of the *degrees of r.e. sets without r-maximal supersets* contains $\mathbf{H}_1$ (Lerman [1971a], Shore), but a complete characterization of it is not known.

Exercise IX.2.33 *Every coinfinite r.e. set has a hypersimple superset.* (Martin [1963]) (Hint: if an r.e. coinfinite set A is not hypersimple, let f be a disjoint strong array intersecting $\overline{A}$ and covering ω, B be a hypersimple set, and consider $A \cup \bigcup_{x \in B} D_{f(x)}$.)

The world of simple sets $\star$

We summarize here the relationships among various types of r.e. sets introduced in Chapter III and in the present one, namely: simple (III.2.9), hypersimple (III.3.7), dense simple (III.3.9), hyperhypersimple (III.4.5), finitely strongly hypersimple (III.4.9), strongly hypersimple (III.4.9), maximal (III.4.13), r-maximal (IX.2.12), and quasimaximal (IX.2.22).

Our goal is to show that *the following implications hold, and no other one does*:

$$\begin{tikzcd}
 & r\text{-max} \arrow[drr] & & & & & \\
\text{max} \arrow[ur] \arrow[dr] & & & \text{shs} \arrow[r] & \text{fshs} \arrow[dr] & & \\
 & q\text{-max} \arrow[r] & \text{hhs} \arrow[ur] \arrow[drr] & & & \text{hs} \arrow[r] & \text{simple} \\
 & & & & \text{dense} \arrow[ur] & &
\end{tikzcd}$$

We have already proved that the implications hold in III.3.9.b, III.4.9, IX.2.13.e, IX.2.22 and IX.2.25. We thus have only to provide counterexamples to the missing implications. They come from different sources, including Post [1944], Dekker [1954], Yates [1962], Martin [1963], [1966a], Young [1966a], Robinson [1967], [1967a], [1968], and Lachlan [1968b].

We first prove that no other arrow holds.

1. *There is an r-maximal, not dense simple set*
   As in IX.2.27 and IX.2.32, notice that if A is dense simple then there is no recursive function h such that if

$$I_n = \{x : h(n) \leq x < h(n+1)\}$$

then, for infinitely many n, $I_n \cap \overline{A}$ has at least n elements: otherwise, for infinitely many n, $h(n+1)$ bounds the n-th element of $\overline{A}$, and the function enumerating $\overline{A}$ by magnitude does not dominate every total recursive function, contradicting the definition of dense simplicity (III.3.9).

We now build A r-maximal and h recursive such that there are infinitely many n such that $I_n \cap \overline{A}$ has at least n elements: then A is not dense simple.

To make A r-maximal, we want to avoid that if $\mathcal{W}_e = \overline{W}_i$ then both $\mathcal{W}_e \cap \overline{A}$ and $\mathcal{W}_i \cap \overline{A}$ are infinite: we achieve this by ensuring that $\overline{\mathcal{W}}_i$ and $\overline{\mathcal{W}}_e$ intersect on $\overline{A}$, whenever both $\overline{\mathcal{W}}_i \cap \overline{A}$ and $\overline{\mathcal{W}}_e \cap \overline{A}$ are infinite.

If $\overline{\mathcal{W}}_e \cap \overline{A}$ is infinite then $I_n \cap \overline{\mathcal{W}}_e \cap \overline{A} \neq \emptyset$ for infinitely many n, and similarly for $\overline{\mathcal{W}}_i$. We would like an n that works for both $\overline{\mathcal{W}}_e$ and $\overline{\mathcal{W}}_i$ simultaneously, which is automatic if we ensure that, if $I_n \cap \overline{\mathcal{W}}_e \cap \overline{A} \neq \emptyset$ holds for infinitely many n, then it holds for almost every n. This is obtained by building A as in the construction of a maximal set (III.4.18), using as e-states the following:

$$\sigma(e, x, s) = \sum \{2^{e-i} : i \leq e \ \wedge \ I_x \cap \overline{A}_s \subseteq \mathcal{W}_{i,s}\}.$$

The construction thus ensures that if both $\overline{\mathcal{W}}_e \cap \overline{A}$ and $\overline{\mathcal{W}}_i \cap \overline{A}$ are infinite then, for almost every n,

$$I_n \cap \overline{\mathcal{W}}_e \cap \overline{A} \neq \emptyset \qquad \text{and} \qquad I_n \cap \overline{\mathcal{W}}_i \cap \overline{A} \neq \emptyset.$$

To make $\overline{\mathcal{W}}_e$ and $\overline{\mathcal{W}}_i$ intersect on $\overline{A}$, it is enough to force each of them to have more than half of the elements of $I_n \cap \overline{A}$, whenever they have some at all: then they must have at least one such element in common.

In the construction of A we thus alternate steps to maximize the e-state of the e-th element of $\overline{A}$, to steps that put the elements of $I_n \cap \overline{\mathcal{W}}_{e,s} \cap \overline{A}_s$ into A, whenever their number has dropped to half (or less) of the number of elements of $I_n \cap \overline{A}_s$.

Since we want A coinfinite, we cannot let every $\overline{\mathcal{W}}_e$ interfere with every I_n, and we thus allow it to do so only for $n \geq e$. Since every $\overline{\mathcal{W}}_e$ contributes at most half of the elements of $I_n \cap \overline{A}_s$ to A, to get $I_n \cap \overline{A} \neq \emptyset$ it is enough to have I_n with at least 2^{n+1} elements (so that, even after $n+1$ halving, at least one remains).

Finally, to have A not dense simple we want $I_n \cap \overline{A}$ not only nonempty, but with at least n elements: since having 2^{n+1} elements in I_n ensures that there is at least one in $I_n \cap \overline{A}$, having $n \cdot 2^{n+1}$ in I_n ensures that there are at least n in $I_n \cap \overline{A}$. We thus let

$$h(0) = 0 \qquad \text{and} \qquad h(n+1) = h(n) + n \cdot 2^{n+1}.$$

2. *There is a quasimaximal, not r-maximal set*
   Since the maximal sets are not closed under intersection (III.4.20), there is a quasimaximal, not maximal set: by IX.2.20, such a set is also not r-maximal.

3. *There is a dense simple, not finitely strongly hypersimple set*
   A first example is obtained by a direct construction. We build an r.e. set A by stages. At stage s we have A_s, and let $\overline{A}_s = \{a_0^s < a_1^s < \cdots\}$. At the end $\overline{A} = \{a_0 < a_1 < \cdots\}$, where $a_n = \lim_{n\to\infty} a_n^s$.

   To have A dense simple we make sure that, for any total recursive function φ_e and almost every n, $\varphi_e(n) < a_n$. This is achieved by ensuring that, for each $e, n \le s$, if $\varphi_{e,s}(n)\downarrow$ then $\varphi_{e,s}(n) < a_n^{s+1}$.

   To have A not finitely strongly hypersimple, we define a weak array $\{\mathcal{W}_{f(n)}\}_{n\in\omega}$ by putting, at stage s and for each $n \le s$, a_n^s into $\mathcal{W}_{f(n)}$. Since the limit of a_n^s exists, each $\mathcal{W}_{f(n)}$ is finite. Since $a_n \in \mathcal{W}_{f(n)}$, each $\mathcal{W}_{f(n)}$ intersects $\overline{A}$, and their union covers $\overline{A}$. To have the $\mathcal{W}_{f(n)}$'s disjoint it is enough to make sure that, if a_n^{s+1} moves at stage $s+1$, then it goes to an element bigger than all those already in $\mathcal{W}_{f(m)}$, for any $m \le s$.

   Alternatively one can note that, by the proof of III.5.7, a semirecursive dense simple set is not finitely strongly hypersimple. It is thus enough to get a dense simple deficiency set. A natural example is exhibited in IX.2.34, but the following direct construction is simpler. We build a total recursive function f by stages, by defining $f(s)$ at stage s, and let

   $$x \in A \Leftrightarrow (\exists y > x)(f(y) \le f(x)).$$

   Let $f(0) = 0$. At stage $s+1$, let

   $$a_0^s < a_1^s < \cdots < a_m^s \le s$$

   be the elements of $\overline{A}_s$ below s. See if there is $e \le s$ such that, for some $e \le n \le m$, $\varphi_{e,s}(n)\downarrow$ and $a_n^s \le \varphi_e(n)$. If so, take the smallest such e, and let $f(s+1) = f(a_n^s)$, so that $a_n^{s+1} = s+1$: since, by convention, if $\varphi_{e,s}(n)\downarrow$ then $\varphi_e(n) \le s$, this ensures that $\varphi_e(n) < a_n^{s+1} \le a_n$. Otherwise, let $f(s+1) = f(a_m^s) + 1$, so that $a_{m+1}^{s+1} = s+1$.

   Since a_n^s moves only for $e \le n$, $a_n = \lim_{n\to\infty} a_n^s$ exists. Moreover, if φ_e is total then $\varphi_e(n) < a_n$ for $e \le n$, and A is dense simple.

We now prove that no arrow can be reversed.

4. *There is a simple, not hypersimple set*
   By the proof of III.3.5, Post's simple set (III.2.11) is not hypersimple.

5. *There is a hypersimple, not finitely strongly hypersimple set*
   By the proof of III.5.7, a semirecursive hypersimple set (e.g. any deficiency set) is not finitely strongly hypersimple.

6. *There is a finitely strongly hypersimple, not strongly hypersimple set*
   By III.4.9.a.

7. *There is a strongly hypersimple, not r-maximal set*
   The set considered in part 2 is strongly hypersimple, because quasimaximal.

8. *There is an r-maximal, not maximal set*
   By IX.2.17, a major subset of a maximal set is r-maximal, but not maximal.

9. *There is a strongly hypersimple, not hyperhypersimple set*
   The set considered in part 8 is strongly hypersimple, because r-maximal, and not hyperhypersimple. Otherwise, being r-maximal, it would be maximal by IX.2.20.

10. *There is a hyperhypersimple, not quasimaximal set*
    Any hyperhypersimple set not contained in a maximal set is obviously not quasimaximal, and such a set exists by IX.2.28.f.

11. *There is a quasimaximal, not maximal set*
    By III.4.20, maximal sets are not closed under intersection.

12. *There is a hypersimple, not dense simple set*
    By III.3.13, every nonrecursive r.e. degree contains a hypersimple set, while by the proof of IX.2.24 only high r.e. degrees contain dense simple sets.

13. *There is a dense simple, not hyperhypersimple set*
    The set considered in part 3 is dense simple, and not hyperhypersimple.

Unfortunately, most of the examples given above are produced by direct *ad hoc* constructions. The next result provides a natural example of a dense simple set, with a number of additional interesting properties.

Proposition IX.2.34 (Robinson [1968]) *The deficiency set of $\mathcal{K}$ is dense simple.*

Proof. Let f be a one-one recursive function with range $\mathcal{K}$, and

$$x \in B \Leftrightarrow (\exists y > x)(f(y) < f(x)).$$

Since the elements of B are called deficiency *stages* of the enumeration, we will call their images via f deficiency *elements* of $\mathcal{K}$, and $f(B)$ deficiency *subset* of $\mathcal{K}$. Similarly for nondeficiency stages and elements.

We consider, as in IX.2.23, the following version of the computation function

$$C_{\mathcal{K}}(x) = \mu s. (\forall y \leq x)(y \in \mathcal{K}_s \Leftrightarrow y \in \mathcal{K}),$$

and obtain the result by a sequence of steps, according to the following intuition. Since the computation function of *some* r.e. set A dominates every recursive function by IX.2.23, the same must be true of $\mathcal{K}$ by m-completeness; any recursive enumeration of $\mathcal{K}$ is thus reluctant, and since the nondeficiency elements of $\mathcal{K}$ are the last ones to be enumerated, they are difficult to separate from those in $\overline{\mathcal{K}}$, that are not enumerated at all; but such a separation could be obtained from any recursive function bounding infinitely many nondeficiency stages.

1. *$C_{\mathcal{K}}$ dominates every total recursive function*
   By IX.2.23, there is an r.e. set A such that the function

   $$C_A(x) = \mu s. (\forall y \leq x)(y \in \mathcal{A}_s \Leftrightarrow y \in \mathcal{A})$$

   dominates every total recursive function. Let g be a recursive function such that

   $$x \in A \Leftrightarrow g(x) \in \mathcal{K}.$$

   We will use the following observation: if s is a stage such that

   $$x \in A_s \Leftrightarrow h(x) \in \mathcal{K}_s,$$

   where

   $$h(x) \geq \max_{z \leq x} g(z),$$

   then

   $$C_{\mathcal{K}}(h(x)) \leq s \Rightarrow C_A(x) \leq C_{\mathcal{K}}(h(x)),$$

   i.e. if $\mathcal{K}$ is correct at stage s on $\{0, \ldots, h(x)\}$, and hence on $g(0), \ldots, g(x)$, then so is A on $\{0, \ldots, x\}$.

   Suppose now that some total recursive function φ_e is not dominated by $C_{\mathcal{K}}$: we want to find a total recursive function t which is not dominated by C_A, contradicting the hypothesis.

   Let z be such that $C_{\mathcal{K}}(z) \leq \varphi_e(z)$. If h is nondecreasing and going to infinity, which we may suppose by defining

   $$h(x) = \max\{x, g(0), \ldots, g(x)\},$$

there is x such that

$$h(x) \leq z \leq h(x+1).$$

Since $h(x) \leq z$, by monotonicity of $C_{\mathcal{K}}$ we have

$$C_{\mathcal{K}}(h(x)) \leq C_{\mathcal{K}}(z) \leq \varphi_e(z).$$

We want

$$C_A(x) \leq C_{\mathcal{K}}(h(x)) \qquad \text{and} \qquad \varphi_e(z) \leq t(x),$$

so that

$$C_A(x) \leq t(x).$$

- $\varphi_e(z) \leq t(x)$ can be ensured directly, since $z \leq h(x+1)$, by letting

  $$t(x) \geq \max_{z \leq h(x+1)} \varphi_e(z).$$

- $C_A(x) \leq C_{\mathcal{K}}(h(x))$ follows from the observation at the beginning, if $t(x)$ is a stage s such that

  $$x \in A_s \Leftrightarrow h(x) \in \mathcal{K}_s$$

  because then, by the first condition on t,

  $$C_{\mathcal{K}}(h(x)) \leq C_{\mathcal{K}}(z) \leq \varphi_e(z) \leq t(x) = s.$$

It is thus enough to let

$$t(x) = \mu s \left[\left(s \geq \max_{z \leq h(x+1)} \varphi_e(z)\right) \wedge \left(x \in A_s \Leftrightarrow h(x) \in \mathcal{K}_s\right) \right].$$

If $C_{\mathcal{K}}$ does not dominate φ_e then $C_{\mathcal{K}}(z) \leq \varphi_e(z)$ for infinitely many z, and hence $C_A(x) \leq t(x)$ infinitely many x, i.e. C_A does not dominate t.

2. *$f(B)$ is a major subset of $\mathcal{K}$*
   Suppose the deficiency subset $f(B)$ of $\mathcal{K}$ is not a major subset of it: then there is an r.e. set C such that

   $$\overline{\mathcal{K}} \subseteq C \qquad \text{and} \qquad f(\overline{B}) \not\subseteq^* C,$$

   i.e. C contains $\overline{\mathcal{K}}$, but leaves out an infinite number of nondeficiency elements of $\mathcal{K}$.

   Consider the set R defined as follows: simultaneously enumerate $\mathcal{K}$ and C, and put x in R if and only if it shows up first in $\mathcal{K}$. R is a recursive subset of $\mathcal{K}$ that contains all elements of $\mathcal{K} - C$, and hence an infinite number of nondeficiency elements.

Now let $h(x)$ be the first stage in which the smallest $z \geq x$ that is in R enters $\mathcal{K}$. Since h is a recursive function, it is enough to show that $C_{\mathcal{K}}$ does not dominate it to reach a contradiction with part 1, and thus prove that $f(B)$ must be a major subset of $\mathcal{K}$.

Take any nondeficiency element $x \in R$. Since x is in R, it is the smallest $z \geq x$ that is in R, i.e. $x \in \mathcal{K}_{h(x)}$; and since x is a nondeficiency element, $\mathcal{K}$ must have settled on $\{0, \ldots, x\}$ by stage $h(x)$, i.e. $C_{\mathcal{K}}(x) \leq h(x)$. But there are infinitely many such x's by the definition of R, and thus $C_{\mathcal{K}}$ does not dominate h.

3. *B is dense simple*
Suppose B is not dense simple: if

$$\overline{B} = \{s_0 < s_1 < \cdots\}$$

are the nondeficiency stages ordered by magnitude, there is then a total recursive function g such that, for infinitely many x,

$$s_x < g(x),$$

i.e. below $g(x)$ there are at least x nondeficiency stages.

Consider the set C defined as follows: x is in C if and only if it has not yet been generated by f at stage $g(2x+1)$. C is r.e. (actually, recursive), and it contains $\overline{\mathcal{K}}$ because the elements of the latter are *never* generated. We show that C leaves out an infinite number of nondeficiency elements of $\mathcal{K}$, contradicting the fact that $f(B)$ is a major subset of $\mathcal{K}$.

By the choice of g, for infinitely many x there are at least $2x+1$ nondeficiency stages below $g(2x+1)$: we show that at most $x+1$ of the corresponding nondeficiency elements are in C, and hence at least x are not in C.

Suppose that $f(s) \in C$, for some $s \leq g(2x+1)$. By definition of C, $f(s)$ has not yet been generated at stage $g(2f(s)+1)$: but since it *has* been generated by stage $g(2x+1)$, and g may be supposed to be nondecreasing,

$$2f(s) + 1 \leq 2x + 1$$

and

$$f(s) \leq x,$$

i.e. at most $x+1$ such elements $f(s)$ can be in C. □

The proof actually shows more than stated, namely: by part 1, *the deficiency set of a wtt-complete set is dense simple* (by using as g the recursive bound of the use of a *wtt*-reduction of A); and by part 2, *the deficiency subset of any set A such that C_A dominates every total recursive function is a major subset of A* (such sets exist in any high r.e. degree, by IX.2.23).

We can now summarize the structural properties of two naturally defined sets, already considered on p. I.349 for their completeness properties.

1. *Post's simple set* (III.2.11) is simple, not hypersimple (by the proof of III.3.5), and without dense simple supersets (by III.3.9.c and III.4.24.d).

2. *The deficiency set of $\mathcal{K}$* is dense simple (by IX.2.34), not finitely strongly hypersimple (by the proof of III.5.7, being semirecursive by III.5.6), and without strongly hypersimple supersets (as in IX.2.18, by iterating the Owings Splitting Theorem). Actually, Stob [1979] and Herrmann [1983a] have proved that the deficiency set of $\mathcal{K}$ does not have finitely strongly hypersimple supersets.

By IX.2.33, every coinfinite r.e. set has a hypersimple superset: in terms of the simple sets we studied, the best that one can get for a coinfinite r.e. set is thus either no dense simple superset, or no finitely strongly hypersimple superset, and we have just exhibited natural examples of the two possibilities.

Finally, we consider the classes of r.e. degrees containing r.e. sets of the types considered above.

1. *An r.e. degree contains a maximal (or an r-maximal, or a quasimaximal, or a hyperhypersimple, or a strongly hypersimple, or a finitely strongly hypersimple, or a dense simple) set if and only if it is high*
   By the proof of IX.2.24, every high r.e. degree contains a maximal set, and every dense simple set has a high degree. It only remains to prove that *a finitely strongly hypersimple set has a high degree*, too: this is done by a modification of the proof that an r-maximal set has a high degree (IX.2.25.c), in which instead of building a pair of recursive sets intersecting $\overline{A}$ infinitely often, one builds a weak array $\{\mathcal{W}_{f(x)}\}_{x\in\omega}$ of r.e. sets intersecting $\overline{A}$, and covering it.

   More precisely, given A not of a high degree, at each stage s we put exactly one element in exactly one of $\mathcal{W}_{f(0)}, \ldots, \mathcal{W}_{f(s)}$, so that there are exactly s elements in $\mathcal{W}_{f(0),s-1}, \ldots, \mathcal{W}_{f(s-1),s-1}$. Let $f(s)$ be the least stage t such that the first $s+1$ elements of $\overline{A}_t$ are in $\overline{A}$. Since f is recursive in A, and A is not of a high degree, by XI.1.10 there is a recursive function g such that $f(s) \leq g(s)$ for infinitely many s. The construction is thus the following: at stage s, take the smallest element not in

$$A_{g(s)} \cup \mathcal{W}_{f(0),s-1} \cup \cdots \cup \mathcal{W}_{f(s-1),s-1}$$

and put it in $\mathcal{W}_{f(x)}$, for the smallest $x \leq s$ such that $\mathcal{W}_{f(x),s} \cap \overline{A}_{g(s)} = \emptyset$. By construction each $\mathcal{W}_{f(x)}$ is finite, because it stops getting elements as soon as it gets one of $\overline{A}$; and each element of $\overline{A}$ goes in some $\mathcal{W}_{f(x)}$ because, by the choice of g, infinitely often at stage s one considers the smallest element of $\overline{A}$ not yet used.

2. *An r.e. degree contains a simple (or a hypersimple) set if and only if it is nonrecursive*
   By III.2.14 and III.3.13.

IX.3 Global Properties of R.E. Sets

We now turn to global properties of $\mathcal{E}$ and $\mathcal{E}^*$, and study problems related to definability, homogeneity and automorphisms. We will give solutions to many of these problems by direct methods, without relying indirectly (as it is often the case for degree structures) on coding tools developed to characterize the complexity of the first-order theory. For this reason, and also because the proof of such a characterization would take us too far afield, we only comment about it without proofs.

The complexity of the theory of r.e. sets ⋆

The proof of IX.3.2, applied to sentences, shows that the first-order theories of $\mathcal{E}$ and $\mathcal{E}^*$ have the same degree (and actually the same isomorphism type). We will thus restrict our attention to the latter.

Harrington has proved that *the first-order theory of $\mathcal{E}^*$ has the same degree (and actually the same isomorphism type) as the theory of First-Order Arithmetic*, and in particular it is undecidable and not axiomatizable (Harrington [1983], Herrmann [1984]).

This result is the analogue of various characterizations of the complexity of first-order theories of degrees (see V.7.3, VI.4.8, X.5.27, X.6.31, XI.5.4, XII.3.2, XIII.1.14, XIV.2.13). However, while in all these cases it is actually possible to define a standard model of Arithmetic without parameters, this is impossible here, because Harrington has proved that *no infinite linear ordering is definable without parameters in $\mathcal{E}^*$* (a recursion-theoretical proof is given in Harrington and Nies [1998], and a model-theoretical one in Hodges and Nies [1998]).

Harrington and Herrmann originally obtained the undecidability of the first-order theory of $\mathcal{E}^*$ by interpreting in it the theory of recursive Boolean pairs, i.e. recursive Boolean algebras with distinguished subalgebras, which is undecidable by Rubin [1976] (Boolean algebras with distinguished ideals would not suffice, because their theory is decidable by Ershov [1964]). The proof is an extension of the representation of recursive Boolean algebras by hyperhypersimple sets (see p. 375), and represents recursive Boolean pairs by hyperhypersimple sets and major subsets of them. In particular, it follows that *the theory of hyperhypersimple sets and their major subsets is undecidable* (the theory of hyperhypersimple sets alone is decidable, see p. 375).

As for the full complexity characterization, Harrington's original proof is a constructive version of VI.4.7. More precisely, the translation of Second-Order Arithmetic into the theory of countable distributive lattices with quantification over arbitrary ideals is replaced by a similar translation of First-Order Arithmetic into the theory of countable Boolean pairs with quantification over arithmetical ideals (Boolean algebras with quantification over ideals would not suffice, because their theory is decidable by Rabin [1969]). The second hypothesis of VI.4.7, requesting the embeddability of all countable distributive lattices as initial segments, is replaced by the representability of all countable Boolean pairs as quotients by suitable ideals of a fixed Boolean pair (consisting of the r.e. splittings and the recursive intersections of any fixed hyperhypersimple sets without maximal supersets, see IX.2.28.f). Finally, the first hypothesis of VI.4.7, providing the definability from parameters of all countable ideals, is replaced by the definability from parameters of all suitable ideals of the given Boolean pair.

A simpler and more direct proof, due to Harrington and Nies [1998], is instead a constructive version of V.7.3, in the style of XI.5.4. More precisely, the integers of a standard model are represented by the atoms of the quotient by a suitable ideal of a fixed Boolean algebra (consisting of the r.e. splittings of any fixed hyperhypersimple set such that $\mathcal{L}^*(A)$ has infinitely many atoms), and sets of integers are represented by the ideals generated by the corresponding atoms. An analogue of XI.5.2, providing definability from parameters of all arithmetical relations on the atoms, allows the definition of a standard model. An analogue of XI.3.18, providing uniform definability from parameters of all Σ_n^0 ideals containing all recursive subsets of A, allows the expression of the fact that the model is standard (since n is arbitrary, and not restricted as in XI.3.18, there is no need here to consider effective models as in XI.5.4).

Nies [199?a] has adapted the latter proof to show that *the first-order theory of any interval of $\boldsymbol{\mathcal{E}}^*$ which is not a Boolean algebra has the same degree (and actually the same isomorphism type) as the theory of First-Order Arithmetic.*

Turning to subclasses of sentences, Lachlan [1968c] has proved that *the two-quantifier theory of $\boldsymbol{\mathcal{E}}^*$ is decidable.* As usual, the truth of two-quantifier questions reduces to problems of extensions of embeddings (see p. I.490): on the one hand, necessary conditions are obtained from the existence of canonical realizations, such as maximal and major subsets; on the other hand, the conditions are shown to be sufficient by the method of the Owings Splitting Theorem. Some improvements have been obtained by Degtev [1978a], Lerman and Soare [1980] and Herrmann [1996], but it is not known whether the three-quantifier theory of $\boldsymbol{\mathcal{E}}^*$ is decidable. Nies [199?b] has proved that *the six-quantifier theory of $\boldsymbol{\mathcal{E}}^*$ is undecidable.*

Absolute definability

The next observation is crucial for definability results.

Proposition IX.3.1 (Lacombe) *An r.e. set is finite if and only if every r.e. subset of it is recursive.*

Proof. Since every finite set is recursive, and every subset of a finite set is finite, one direction is immediate.

Conversely, if A is r.e. and infinite, there is a one-one recursive function f with range A (II.1.18). Consider $B = f(\mathcal{K})$. B is an r.e. subset of A, but it is not recursive (otherwise $\overline{\mathcal{K}} = f^{-1}(\overline{B})$ would be r.e.). $\Box$

First we connect definability on $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{E}}^*$.

Proposition IX.3.2 (Lachlan [1968b]) *A property of r.e. sets well-defined on $\boldsymbol{\mathcal{E}}^*$ (i.e. invariant under finite differences) is definable in $\boldsymbol{\mathcal{E}}^*$ if and only if it is definable in $\boldsymbol{\mathcal{E}}$.*

Proof. The direction from $\boldsymbol{\mathcal{E}}^*$ to $\boldsymbol{\mathcal{E}}$ is trivial since, after IX.3.1, we can define $\subseteq^*$ in $\boldsymbol{\mathcal{E}}$:

$$A \subseteq^* B \ \Leftrightarrow\ (\exists F)[F \text{ finite} \wedge A \subseteq B \cup F].$$

For the other direction, suppose $\varphi(X)$ is a formula in the language of $\subseteq$ invariant under finite differences: we want to find a formula $\varphi^*(X)$ in the language of $\subseteq^*$ such that

$$\mathcal{E} \models \varphi(X) \ \Leftrightarrow\ \mathcal{E} \models \varphi^*(X).$$

The idea of the proof is the following: first φ is analyzed into assertions about the finite cardinality of the sets mentioned in it on the one hand, and about $\subseteq^*$ on the other; then the cardinality assertions are replaced by finiteness assertions; finally, an appeal to invariance under finite differences is made, to deduce that the replacement does not affect the truth-value of the formula.

By the usual techniques to reduce a formula to prenex normal form, we can restrict attention to formulas φ of the form

$$(Q_1X_1)\cdots(Q_nX_n)(\exists Y)\ \psi(X, X_1, \ldots, X_n, Y),$$

where ψ is quantifier-free, and each Q_i is either $\exists$ or $\neg\exists\neg$.

1. *cardinality content of quantifier-free formulas*
A trivial, but basic observation is that the atomic statement $X_1 \subseteq X_2$ is equivalent to $X_1 \cap \overline{X_2} = \emptyset$, and hence to $|X_1 \cap \overline{X_2}| = 0$.

We thus proceed by induction, and show how to replace quantifier-free formulas in the variables $X_1, \ldots, X_n$ by cardinality assertions of the form

$$|\tilde{X}_1 \cap \cdots \cap \tilde{X}_n| = m \qquad \text{or} \qquad |\tilde{X}_1 \cap \cdots \cap \tilde{X}_n| \geq m \qquad \text{or} \qquad \textbf{false},$$

where each $\tilde{X}_i$ is either X_i or $\overline{X}_i$. For simplicity, in the following we write $\tilde{\mathcal{X}}$ for $\tilde{X}_1 \cap \cdots \cap \tilde{X}_n$.

Negations are reduced by noting that

$$\begin{array}{lcl} \neg(|\tilde{\mathcal{X}}| = 0) & \Leftrightarrow & |\tilde{\mathcal{X}}| \geq 1 \\ \neg(|\tilde{\mathcal{X}}| = m+1) & \Leftrightarrow & |\tilde{\mathcal{X}}| = 0 \ \vee \cdots \vee \ |\tilde{\mathcal{X}}| = m \ \vee \ |\tilde{\mathcal{X}}| \geq m+2 \\ \neg(|\tilde{\mathcal{X}}| \geq 0) & \Leftrightarrow & \textbf{false} \\ \neg(|\tilde{\mathcal{X}}| \geq m+1) & \Leftrightarrow & |\tilde{\mathcal{X}}| = 0 \ \vee \cdots \vee \ |\tilde{\mathcal{X}}| = m \\ \neg\, \textbf{false} & \Leftrightarrow & |\tilde{\mathcal{X}}| \geq 0. \end{array}$$

Conjunctions are reduced by noting that

$$\begin{array}{lcl} |\tilde{\mathcal{X}}| = m_1 \ \wedge \ |\tilde{\mathcal{X}}| = m_2 & \Leftrightarrow & \begin{cases} |\tilde{\mathcal{X}}| = m_1 & \text{if } m_1 = m_2 \\ \textbf{false} & \text{otherwise} \end{cases} \\ |\tilde{\mathcal{X}}| \geq m_1 \ \wedge \ |\tilde{\mathcal{X}}| = m_2 & \Leftrightarrow & \begin{cases} |\tilde{\mathcal{X}}| = m_2 & \text{if } m_2 \geq m_1 \\ \textbf{false} & \text{otherwise} \end{cases} \\ |\tilde{\mathcal{X}}| \geq m_1 \ \wedge \ |\tilde{\mathcal{X}}| \geq m_2 & \Leftrightarrow & |\tilde{\mathcal{X}}| \geq \max\{m_1, m_2\}. \end{array}$$

Moreover, a conjunction one of whose conjuncts is **false** is equivalent to **false**.

By using the standard techniques to obtain disjunctive normal forms, one can then reduce any quantifier-free formula ψ in the language of $\subseteq$ to a disjunction

$$\bigvee \psi_i,$$

where each ψ_i is a conjunction of atomic cardinality statements of the three forms considered above (by the reduction rules for conjunction, each cardinality statement is about a different intersection $\tilde{X}_1 \cap \cdots \cap \tilde{X}_n$).

2. *cardinality content of existential formulas*
Since

$$(\exists Y)(\bigvee \psi_i) \ \Leftrightarrow \ \bigvee (\exists Y \psi_i),$$

it is enough to consider existential quantifications of conjunctions of atomic cardinality statements of the three forms considered above.

The case

$$(\exists Y)(\textbf{false}) \;\Leftrightarrow\; \textbf{false}$$

is trivial, and the following are easy:

$$\begin{array}{lcl}(\exists Y)(|\tilde{\mathcal{X}} \cap Y| = m_1 \;\wedge\; |\tilde{\mathcal{X}} \cap \overline{Y}| = m_2) & \Leftrightarrow & |\tilde{\mathcal{X}}| = m_1 + m_2 \\ (\exists Y)(|\tilde{\mathcal{X}} \cap Y| \geq m_1 \;\wedge\; |\tilde{\mathcal{X}} \cap \overline{Y}| = m_2) & \Leftrightarrow & |\tilde{\mathcal{X}}| \geq m_1 + m_2 \\ (\exists Y)(|\tilde{\mathcal{X}} \cap Y| \geq m_1 \;\wedge\; |\tilde{\mathcal{X}} \cap \overline{Y}| \geq m_2) & \Leftrightarrow & |\tilde{\mathcal{X}}| \geq m_1 + m_2,\end{array}$$

where missing assertions of the form $\tilde{\mathcal{X}} \cap \tilde{Y}$ can be filled in by adding redundant assertions of the form $|\tilde{\mathcal{X}} \cap \tilde{Y}| \geq 0$.

If these were all possible cases, the previous procedure would replace each existential formula φ by an equivalent quantifier-free formula involving only cardinality statements. The problem is that ψ_i may involve cardinality assertions about intersections $\tilde{\mathcal{X}} \cap \tilde{Y}$ for different $\tilde{\mathcal{X}}$'s and the same Y, and this produces a complication.

To illustrate the problem, we first show e.g. that

$$\begin{array}{l}(\exists Y)(|\tilde{\mathcal{X}}_1 \cap Y| = m_1 \wedge |\tilde{\mathcal{X}}_2 \cap \overline{Y}| = m_2) \;\Leftrightarrow \\ \qquad |\tilde{\mathcal{X}}_1| \geq m_1 \wedge |\tilde{\mathcal{X}}_2| \geq m_2 \wedge (\exists Y)(|\tilde{\mathcal{X}}_1 \cap Y| =^* \emptyset \wedge |\tilde{\mathcal{X}}_2 \cap \overline{Y}| =^* \emptyset).\end{array}$$

The left-to-right direction is obvious, since the cardinality assertions on the right have been obtained by the previous procedure, while the quantified formula on the right is a weakening of the formula on the left, obtained by replacing specific cardinality assertions by generic finiteness assertions.

For the right-to-left direction, suppose Y satisfies the quantified formula on the right. Notice that $\tilde{\mathcal{X}}_1$ and $\tilde{\mathcal{X}}_2$ are disjoint, being by definition different intersections of the form $\tilde{X}_1 \cap \cdots \cap \tilde{X}_n$: if $|\tilde{\mathcal{X}}_1 \cap Y| > m_1$ we can thus take the appropriate number of elements of $\tilde{\mathcal{X}}_1$ out of Y without interfering with $\tilde{\mathcal{X}}_2 \cap \overline{Y}$, in particular by keeping its cardinality unchanged; while if $|\tilde{\mathcal{X}}_1 \cap Y| < m_1$ we can instead put the appropriate number of elements of $\tilde{\mathcal{X}}_1$ into Y, which is possible because $|\tilde{\mathcal{X}}_1| \geq m_1$, again without interfering with $\tilde{\mathcal{X}}_2 \cap \overline{Y}$. Similarly, if $|\tilde{\mathcal{X}}_2 \cap \overline{Y}| \neq m_2$ we can either put some elements of $\tilde{\mathcal{X}}_2$ into Y or take some out, using the hypothesis $|\tilde{\mathcal{X}}_2| \geq m_2$, without interfering with $\tilde{\mathcal{X}}_1 \cap Y$. After these finite changes Y will have the required number of elements, and will thus satisfy the formula on the left-hand-side.

In general, if ψ_i is a conjunction of atomic cardinality statements then

$$(\exists Y)\psi_i \Leftrightarrow \chi_i \wedge (\exists Y)\psi_i^*,$$

where χ_i is a conjunction of atomic cardinality statements obtained as in the first part of the proof, and ψ_i^* is obtained from ψ_i as in the second part of the proof, i.e. as follows:

- each $|\tilde{\mathcal{X}}| = m$ is replaced by $\tilde{\mathcal{X}} =^* \emptyset$
- each $|\tilde{\mathcal{X}}| \geq m$ is replaced by the identically true formula $\emptyset =^* \emptyset$
- **false** is replaced by the identically false formula $\emptyset \neq^* \emptyset$.

The first case thus replaces specific finite cardinality assertions by generic finiteness assertions, while the second case disregards assertions about finite cardinality lower bounds: each ψ_i is thus reduced to a weaker formula ψ_i^* in the language of $\subseteq^*$.[1]

3. *definition of φ^**
   Given a formula φ in prenex normal form

   $$(Q_1X_1)\cdots(Q_nX_n)(\exists Y)\ \psi,$$

   where ψ is quantifier-free, the previous two steps show how to reduce

   $$(\exists Y)\psi$$

   to a formula of the form

   $$\bigvee(\psi_i \wedge \eta_i),$$

   where ψ_i is a conjunction of cardinality statements and η_i is a formula in the language of $\subseteq^*$.

   It remains to deal with the rest of the quantifiers, which can be done inductively by showing that negations and existential quantifications of formulas of the form above are equivalent to formulas of the same form.

   Negation is treated by noting that

   $$\neg\bigvee(\psi_i \wedge \eta_i) \Leftrightarrow \bigwedge(\neg\psi_i \vee \neg\eta_i),$$

   and that $\neg\psi_i$ is equivalent to a disjunction of cardinality assertions by the first part of the proof, while $\neg\eta_i$ is still a formula in the language of $\subseteq^*$. It is then enough to distribute conjunctions over disjunctions.

[1] If one wants to avoid mentioning $\emptyset$, then $\tilde{\mathcal{X}} =^* \emptyset$ can be replaced by $(\forall Y)(\tilde{\mathcal{X}} \subseteq^* Y)$, and similarly in the other cases.

Existential quantification is treated by noting that

$$\exists Y \bigvee (\psi_i \wedge \eta_i) \;\Leftrightarrow\; \bigvee \exists Y (\psi_i \wedge \eta_i),$$

and that

$$\exists Y (\psi_i \wedge \eta_i) \;\Leftrightarrow\; \chi_i \wedge \exists Y (\psi_i^* \wedge \eta_i)$$

by the second part of the proof.

After eliminating all quantifiers, the formula $\varphi(X)$ is thus reduced to an equivalent one of the form

$$\bigvee (\psi_i \wedge \eta_i),$$

where the only set variable appearing in ψ_i is X. We finally define φ^* to be

$$\bigvee (\psi_i^* \wedge \eta_i),$$

which is now a formula in the language of $\subseteq^*$.

Obviously, if φ holds in $\mathcal{E}$ then so does φ^*. Conversely, if φ^* holds in $\mathcal{E}$ then so does φ: indeed, since the latter is invariant under finite differences, only generic finiteness assertions are relevant in it, while specific cardinalities or lower bounds are redundant, and can thus be safely disregarded, which is precisely what φ^* does. $\Box$

We now produce a number of definability results.

Proposition IX.3.3 Definability in $\mathcal{E}$ (Lacombe, Lachlan [1968b]) *The following properties of sets are definable in $\mathcal{E}$: being finite, infinite, recursive, simple, hyperhypersimple, r-maximal, or maximal.*

Proof. In the language of inclusion one can trivially define union, intersection, $\emptyset$ and ω. IX.3.1 allows us to talk about finite and cofinite sets:

$$\begin{array}{lcl} A \text{ recursive} & \Leftrightarrow & (\exists B)(A \cap B = \emptyset \;\wedge\; A \cup B = \omega) \\ A \text{ finite} & \Leftrightarrow & (\forall B)(B \subseteq A \;\rightarrow\; A \text{ recursive}) \\ A \text{ cofinite} & \Leftrightarrow & (\exists F \text{ finite})(A \cup F = \omega). \end{array}$$

We can now translate the definitions of the stated notions, the only nontrivial case being hyperhypersimplicity (that requires IX.2.10). For example, for coinfinite sets:

$$\begin{array}{lcl} A \text{ simple} & \Leftrightarrow & (\forall B)(B \cap A = \emptyset \;\rightarrow\; B \text{ finite}) \\ A \text{ hyperhypersimple} & \Leftrightarrow & (\forall B \supseteq A)(\exists C \supseteq A)(B \cap C = A \;\wedge\; B \cup C = \omega) \\ A \text{ maximal} & \Leftrightarrow & (\forall B \supseteq A)(\exists F \text{ finite})(A \cup F = B \;\vee\; B \cup F = \omega). \end{array}$$

To avoid a misunderstanding, note that all quantifiers on the right-hand-side range over r.e. sets. Also, the slight awkwardness of some definitions is due to the fact that we cannot talk of complements of r.e. sets directly. $\Box$

It follows from IX.3.3 that the properties of (not) having maximal, hyperhypersimple, or r-maximal supersets are also definable.

Exercises IX.3.4 a) *The ideal of finite sets is the only nontrivial definable ideal of* $\boldsymbol{\mathcal{E}}$. (Lerman [1976]) (Hint: as in IX.1.9.1.)

b) *The filter of cofinite sets is the smallest nontrivial definable filter of* $\boldsymbol{\mathcal{E}}$. (Lerman [1976]) (Hint: dually to part a), a nontrivial definable filter contains all cofinite sets.)

c) *The filter of simple or cofinite sets is the largest nontrivial definable filter of* $\boldsymbol{\mathcal{E}}$. (Lerman [1976]) (Hint: suppose A belongs to a definable filter, and is not simple. Then $\overline{A}$ contains an infinite r.e. set, and hence an infinite and coinfinite recursive set B. Since B and $\overline{B}$ are automorphic, as in part a), the filter is trivial.)

d) *The hypersimple or cofinite sets are a filter, but not a definable one; the hyperhypersimple or cofinite sets are a definable filter; the maximal or cofinite sets are a definable class but not a filter*. (Hint: see III.3.11.b, IX.3.24, III.4.11, III.4.20.a.)

The proposition above takes care of almost all the major species of r.e. sets introduced so far, and it only leaves out the hypersimple and creative sets. We dispose of the latter now, and defer the treatment of the former to IX.3.24.

Theorem IX.3.5 Lattice-Theoretical Characterization of Creativity (Harrington) *The following are equivalent, for any r.e. set A:*

1. *A is creative*
2. $(\exists B)(\forall C)(\exists R \text{ recursive})(R \cap B \text{ not recursive} \wedge R \cap A = R \cap C)$, *where all quantifiers range over r.e. sets.*

Proof. The idea for the proof comes from the observation that the set A defined by

$$\langle x, e \rangle \in A \Leftrightarrow \langle x, e \rangle \in \mathcal{W}_e$$

is creative: it is enough to fix the set $\mathcal{K} \cdot N$, and to consider any index e such that $\mathcal{W}_e = \mathcal{K} \cdot N$, to have

$$x \in \mathcal{K} \Leftrightarrow \langle x, e \rangle \in \mathcal{W}_e \Leftrightarrow \langle x, e \rangle \in A.$$

We have thus been able to obtain an m-reduction of $\mathcal{K}$ to A from the following lattice-theoretical fact about r.e. sets:

$$(\exists B)(\forall C)(\exists R \text{ infinite})(R \cap A = R \cap C),$$

where $B = \mathcal{K} \cdot N$, $C = \mathcal{W}_e$, and $R = N \cdot \{e\}$, so that the condition

$$R \cap A = R \cap C$$

just expresses the definition of A, i.e.

$$\langle x, e \rangle \in A \;\Leftrightarrow\; \langle x, e \rangle \in \mathcal{W}_e.$$

We can now try to define an m-reduction of $\mathcal{K}$ to A, for any r.e. set A satisfying the previous lattice-theoretical property. We fix B, which is given. For any C we know that there is an infinite r.e. set R such that $R \cap A = R \cap C$: if we build C such that R is the range of an m-reduction of $\mathcal{K}$ to C, we then automatically get an m-reduction of $\mathcal{K}$ to A.

If we could use R in the construction of C, we could simply one-one enumerate the elements of R, and put the n-th one, x_n, into C_s if and only if $n \in \mathcal{K}_s$. Then at the end

$$n \in \mathcal{K} \;\Leftrightarrow\; x_n \in C \;\Leftrightarrow\; x_n \in R \cap C \;\Leftrightarrow\; x_n \in R \cap A \;\Leftrightarrow\; x_n \in A,$$

because the x_n's are all distinct, $x_n \in R$ by definition, and $R \cap C = R \cap A$.

To know R before hand, we would like to appeal to the Fixed-Point Theorem: this we could do if we knew that R is obtained effectively from C, as it is in the previous example, but the problem is that we cannot express this effectiveness requirement in lattice-theoretical terms. We thus have to use all possible R's in the construction of C, knowing that one of them will work at the end.

The construction of C is now modified as follows. For any e, we let x_n^e be the n-th element in a one-one enumeration of $\mathcal{W}_e$ (obviously, x_n^e will be defined for all n only if $\mathcal{W}_e$ is infinite). For every e we would like to ensure

$$n \in \mathcal{K} \;\Leftrightarrow\; x_n^e \in C,$$

but the attempts to put x_n^e into C_s if and only if $n \in \mathcal{K}_s$ may conflict, since it is possible that $x_n^e = x_m^i$ for some n, m, and $e \neq i$.

The obvious solution, of giving priority to the smaller index when a conflict arises, here does not make sense: although some $\mathcal{W}_e$ will be the appropriate R for the C we are constructing, we know nothing about e, and in particular there is no guarantee that the smallest one will be the right one. The solution comes from the example above, where the sets R were all disjoint, and thus did not interfere with each other: although we cannot directly express the disjointness condition for infinitely many sets in lattice-theoretical terms, we can express the weaker condition that there is a fixed r.e. set intersecting each R infinitely often, and use its elements for all the codings of $\mathcal{K}$.

We can actually use the set B for this purpose, and require R to be a recursive set such that $R \cap B$ is not recursive. If only finitely elements entered B after having entered R, then $R \cap B$ would be recursive: indeed, $R \cap B$ is r.e., and its complement $\overline{R} \cup \overline{B}$ would only differ finitely from $\overline{R} \cup (R \setminus B)$. Then it must be the case that infinitely many elements enter B only after having entered R, and we can use them for the coding: they cannot equal any x_m^i already defined, because such an x_m^i must already be in B.

We now want to prove that if

$$(\exists B)(\forall C)(\exists R \text{ recursive})(R \cap B \text{ not recursive } \wedge\ R \cap A = R \cap C),$$

then A is creative (the converse is satisfied by the creative set considered at the beginning, because $R = N \cdot \{e\}$ is recursive but $R \cap B = \mathcal{K} \cdot \{e\}$ is not, and hence by all creative sets, by III.7.14).

The construction of C is finally modified as follows. Fix a one-one enumeration of B. At stage s, if z is the new element generated in B, let $z = x_n^e$ for the smallest e (if it exists) such that:

- $z \in \mathcal{W}_{e,s}$
- x_n^e is still undefined
- $(\forall m < n)(x_m^e$ has already been defined)
- $(\forall m < n)(m \in \mathcal{K}_s \ \Leftrightarrow\ x_m^e \in A_s)$.

Moreover, for every e and n such that x_n^e has been defined, let

$$x_n^e \in C_s \ \Leftrightarrow\ n \in \mathcal{K}_s.$$

Notice that if, for some e, x_n^e is eventually defined for every n, then it must be the case that

$$(\forall n)(n \in \mathcal{K} \ \Leftrightarrow\ x_n^e \in A).$$

Indeed, if for some m

$$m \in \mathcal{K} \ \Leftrightarrow\ x_m^e \in A$$

failed at the end, then

$$m \in \mathcal{K}_s \ \Leftrightarrow\ x_m^e \in A_s$$

would fail for every stage s from a certain s_0 on: by construction, after s_0 no new x_n^e could then be defined.

It is thus enough to show that, for some e, x_n^e is defined for every n. Let e be the smallest index such that $\mathcal{W}_e = R$, where R is the set corresponding to the C constructed above, and suppose that only finitely many x_n^i are ever defined for $i < e$ (otherwise there is nothing to prove), and all before a given stage s_0. Since infinitely many elements are generated first in $\mathcal{W}_e$ and then in B, there are infinitely many stages after s_0 in which an element z already generated in $\mathcal{W}_e$ is generated in B, and it can be used to define a new x_n^e, when all the x_m^e for $m < n$ have already been defined.

A crucial observation is that, for every $m < n$, the construction ensures that

$$m \in \mathcal{K} \;\Leftrightarrow\; x_m^e \in C.$$

But, since $\mathcal{W}_e \cap A = \mathcal{W}_e \cap C$ by the choice of e,

$$m \in \mathcal{K} \;\Leftrightarrow\; x_m^e \in A.$$

At any stage $s \geq s_0$ as above such that, for every $m < n$,

$$m \in \mathcal{K}_s \;\Leftrightarrow\; x_m^e \in A_s,$$

one can thus define x_n^e, since e has highest priority. $\square$

Exercise IX.3.6 Lattice-Theoretical Characterization of Effective Inseparability. (Nies) *The following are equivalent, for any pair of disjoint r.e. sets A_0 and A_1:*

1. *A_0 and A_1 are effectively inseparable*
2. *$(\exists B)(\forall C_0, C_1$ disjoint$)(\exists R$ recursive$)(R \cap B$ not recursive $\wedge$ $R \cap A_i = R \cap C_i$ for $i = 0, 1)$, where all quantifiers range over r.e. sets.*

(Hint: in one direction, let $\{(\widehat{\mathcal{W}}_{e_0}, \widehat{\mathcal{W}}_{e_1})\}_{e \in \omega}$ be a recursive enumeration of the disjoint pairs of r.e. sets, and consider the effectively inseparable pair (A_0, A_1) defined by:

$$\langle x, \langle e_0, e_1 \rangle \rangle \in A_i \;\Leftrightarrow\; \langle x, \langle e_0, e_1 \rangle \rangle \in \widehat{\mathcal{W}}_{e_i}.$$

As above, one can let $B = \mathcal{K} \cdot N$, $(C_0, C_1) = (\widehat{\mathcal{W}}_{e_0}, \widehat{\mathcal{W}}_{e_1})$, and $R = N \cdot \{\langle e_0, e_1 \rangle\}$.

The other direction is similar to that above.)

As a further example of a positive result, Harrington and Nies [1998] have proved that *quasimaximality is definable*.

Homogeneity

The principal ideal generated by a finite r.e. set is trivial, since it simply consists of all its subsets. The following observation takes care of the remaining cases.

Proposition IX.3.7 Downward Strong Homogeneity for $\boldsymbol{\mathcal{E}}$. *If A is an infinite r.e. set, then the principal ideal generated by it is isomorphic to $\boldsymbol{\mathcal{E}}$.*

Proof. Let f be a recursive one-one function with range A. Then the map

$$\mathcal{W}_e \longmapsto f(\mathcal{W}_e)$$

is the required isomorphism. $\square$

Since principal ideals are trivial, we turn our attention to principal filters. As the next exercises show, they subsume many other natural substructures of $\boldsymbol{\mathcal{E}}$.

Exercises IX.3.8 a) *The intervals of $\boldsymbol{\mathcal{E}}$ are isomorphic to principal filters.* (Hint: if $A \subseteq B$ and $B - A$ is infinite, let f be a recursive one-one function with range B, and consider $f^{-1}(A)$.)

b) *For any r.e. set $\boldsymbol{A}$, the principal filter $\boldsymbol{\mathcal{L}(A)}$ is isomorphic to the lattice $\boldsymbol{\mathcal{E}(\overline{A})}$ of the r.e. sets restricted to $\boldsymbol{\overline{A}}$.* (Hint: send $\mathcal{W}_e \cup A$ to $\mathcal{W}_e \cap \overline{A}$.)

The next result shows that there are different principal filters.

Proposition IX.3.9 Failure of Upward Homogeneity for $\boldsymbol{\mathcal{E}}$. *If A is hyperhypersimple, then the principal filter $\mathcal{L}(A)$ generated by it is not elementarily equivalent to $\boldsymbol{\mathcal{E}}$.*

Proof. If A is hyperhypersimple, then $\mathcal{L}(A)$ is a Boolean algebra by IX.2.10. But $\boldsymbol{\mathcal{E}}$ is not a Boolean algebra, since any r.e. nonrecursive set is not complemented in it. $\square$

The only principal filters whose isomorphism type can be less complicated than $\boldsymbol{\mathcal{E}}$ itself are those just considered.

Proposition IX.3.10 *If A is a nonhyperhypersimple r.e. set, then $\boldsymbol{\mathcal{E}}$ is embedded in the principal filter $\mathcal{L}(A)$ generated by A.*

Proof. Let $\{\mathcal{W}_{f(x)}\}_{x\in\omega}$ be a disjoint weak array intersecting $\overline{A}$. Then the map

$$\mathcal{W}_e \longmapsto A \cup \Big(\bigcup_{x \in \mathcal{W}_e} \mathcal{W}_{f(x)} \Big)$$

is a one-one map from $\boldsymbol{\mathcal{E}}$ into $\mathcal{L}(A)$ which preserves inclusion. $\square$

A full solution to the upward homogeneity problem would require a complete determination of the r.e. sets A such that $\boldsymbol{\mathcal{E}}$ *is isomorphic to* $\mathcal{L}(A)$, rather than only embedded in it. It is so for all low$_2$ coinfinite r.e. sets (Soare [1982], Harrington, Lachlan, Maass and Soare [199?]), but not for all low$_3$ (since there are such sets without maximal supersets, by IX.2.29). On the other hand, r.e. sets A such that $\boldsymbol{\mathcal{E}}$ is isomorphic to $\mathcal{L}(A)$ exist in every r.e. degree (Soare [1982]).

Maass [1983] has completely determined the coinfinite r.e. sets A such that *$\boldsymbol{\mathcal{E}}$ is effectively isomorphic to $\mathcal{L}(A)$*, i.e. such that the isomorphism is actually induced by a recursive function. More precisely, *$\boldsymbol{\mathcal{E}}$ is effectively isomorphic to $\mathcal{L}(A)$ if and only if $\overline{A}$ is semilow$_{1.5}$* (see IX.4.28). The condition is obviously necessary, since if $\mathcal{L}(A)$ is effectively isomorphic to $\boldsymbol{\mathcal{E}}$, then so is $\boldsymbol{\mathcal{E}}(\overline{A})$ by IX.3.8.b, i.e. there is a recursive function f such that $\Phi(\mathcal{W}_e \cap \overline{A}) = \mathcal{W}_{f(e)}$ induces an automorphism. In particular,

$$\mathcal{W}_e \cap \overline{A} \text{ finite } \Leftrightarrow \mathcal{W}_{f(e)} \text{ finite,}$$

hence $F_{\overline{A}} \leq_1 F \leq_1 \mathcal{K}'$, and $\overline{A}$ is semilow$_{1.5}$.

On can also look at 'external' homogeneity questions, by considering the lattice $\boldsymbol{\mathcal{E}}^{\boldsymbol{A}}$ of sets r.e. in a given set A. Lachlan [1968b] has proved, by a relativization of the characterization of the possible isomorphism types of the principal filters generated by the hyperhypersimple sets (see p. 375), that there exists a set A such that $\boldsymbol{\mathcal{E}}^{\boldsymbol{A}}$ is not isomorphic to $\boldsymbol{\mathcal{E}}$. More generally:

- *if $\boldsymbol{\mathcal{E}}^{\boldsymbol{A}}$ is elementarily equivalent to $\boldsymbol{\mathcal{E}}$, then A is arithmetical* (Harrington and Nies [1998])
- *if $\boldsymbol{\mathcal{E}}^{\boldsymbol{A}}$ is isomorphic to $\boldsymbol{\mathcal{E}}^{\boldsymbol{B}}$, then A and B are arithmetically equivalent* (Hammond [1990]).

In the opposite direction:

- *if A and B have the same double jump, then $\boldsymbol{\mathcal{E}}^{\boldsymbol{A}}$ is isomorphic to $\boldsymbol{\mathcal{E}}^{\boldsymbol{B}}$* (Hammond [1990], Harrington).

Hammond [1990] has proved that *$\boldsymbol{\mathcal{E}}^{\boldsymbol{A}}$ is effectively isomorphic to $\boldsymbol{\mathcal{E}}$ if and only if A is low.* The condition is obviously necessary, since if $\boldsymbol{\mathcal{E}}^{\boldsymbol{A}}$ is effectively isomorphic to $\boldsymbol{\mathcal{E}}$, then there is a recursive function f such that $\Phi(\mathcal{W}_e^A) = \mathcal{W}_{f(e)}$ induces an automorphism. In particular,

$$\mathcal{W}_e^A \text{ finite } \Leftrightarrow \mathcal{W}_{f(e)} \text{ finite,}$$

hence $\Sigma_2^{0,A} = \Sigma_2^0$ by X.9.6, and A is low.

A particularly interesting relativization is obtained for $A = \emptyset^{(n)}$, in which case Post's Theorem (IV.1.14) implies that $\boldsymbol{\mathcal{E}}^{\boldsymbol{A}}$ is the lattice of Σ_{n+1}^0 sets, ordered by inclusion. Harrington and Nies [1998] have proved that *for any $n \neq m$, the lattices of Σ_n^0 and Σ_m^0 sets (modulo finite sets) are not elementarily equivalent*, and the result actually extends to all levels of the Hyperarithmetical Hierarchy (see p. I.391).

Automorphisms

The next definition introduces a standard algebraic notion.

Definition IX.3.11 *A function* $\Phi : \boldsymbol{\mathcal{E}} \to \boldsymbol{\mathcal{E}}$ *is an* **automorphism of** $\boldsymbol{\mathcal{E}}$ *if it is one-one, onto, and it preserves inclusion. Similarly for* **automorphisms of** $\boldsymbol{\mathcal{E}}^*$.

A first way of describing an automorphism consists of specifying the local action on the natural numbers that induces it.

Definition IX.3.12 *A permutation* p **induces** *an automorphism* Φ *of* $\boldsymbol{\mathcal{E}}$ *if, for all* e,

$$\Phi(\mathcal{W}_e) = p(\mathcal{W}_e).$$

Similarly, p *induces an automorphism* Φ *of* $\boldsymbol{\mathcal{E}}^*$ *if, for all* e,

$$\Phi(\mathcal{W}_e^*) = (p(\mathcal{W}_e))^*.$$

A second way of describing an automorphism consists of specifying the global action on the (indices of) r.e. sets that defines it.

Definition IX.3.13 *A function* h **presents** *(or is a* **presentation** *of) an automorphism* Φ *of* $\boldsymbol{\mathcal{E}}$ *if, for all* e,

$$\Phi(\mathcal{W}_e) = \mathcal{W}_{h(e)}.$$

Similarly, h *presents an automorphism* Φ *of* $\boldsymbol{\mathcal{E}}^*$ *if, for all* e,

$$\Phi(\mathcal{W}_e^*) = \mathcal{W}_{h(e)}^*.$$

Notice that every automorphism Φ of $\boldsymbol{\mathcal{E}}$ can be presented by a permutation h: it is enough to define

$$\begin{array}{rcl} h(0) & = & \mu e.\,[\mathcal{W}_e = \Phi(\mathcal{W}_0)] \\ h(n+1) & = & \mu e.\,[\mathcal{W}_e = \Phi(\mathcal{W}_{n+1}) \land e \notin \{h(0), \ldots, h(n)\}]. \end{array}$$

Then h is total because every r.e. set has infinitely many indices, is one-one by definition, and is onto because for every e there are infinitely many n such that $\mathcal{W}_e = \Phi(\mathcal{W}_n)$.

Exercises IX.3.14 Automorphisms of $\boldsymbol{\mathcal{E}}$.

a) *There are* $2^{\aleph_0}$ *automorphisms of* $\boldsymbol{\mathcal{E}}$. (Kent) (Hint: let A be a cohesive set, see III.4.13, and let p be a permutation which is the identity on $\overline{A}$. By cohesiveness, for any r.e. set B both $p(B)$ and $p^{-1}(B)$ are r.e. There are $2^{\aleph_0}$ such permutations p.)

b) *Every automorphism of $\mathcal{E}$ is induced by a permutation of ω.* (Hint: an automorphism is completely determined by its behavior on singletons, which are sent into singletons. Then let $p(x)$ be the only element of $\Phi(\{x\})$.)

c) *Every recursive permutation induces an automorphism of $\mathcal{E}$, but not every automorphism is induced by a recursive permutation.* (Hint: see part a).)

d) *An automorphism of $\mathcal{E}$ is recursively presentable if and only if it is induced by a recursive permutation.* (Hint: one direction comes from the S^m_n-Theorem, since if p is recursive, then there is h recursive such that $p(\mathcal{W}_x) = \mathcal{W}_{h(x)}$. Conversely, let g be a recursive function such that $\mathcal{W}_{g(x)} = \{x\}$, and let $p(x)$ be the only element of $\Phi(\mathcal{W}_{g(x)}) = \mathcal{W}_{h(g(x))}$.)

Exercises IX.3.15 Automorphisms of $\mathcal{E}^*$.

a) *Every automorphism of $\mathcal{E}$ induces an automorphism of $\mathcal{E}^*$.* (Hint: if $\Phi(A) = B$ and $C =^* A$, then $\Phi(C) =^* B$.)

b) *The $2^{\aleph_0}$ automorphisms of IX.3.14.a all induce the same, trivial automorphism of $\mathcal{E}^*$.* (Hint: if p is as in IX.3.14.a, then $p(B) =^* B$ for every r.e. set B, and thus p induces the identity on $\mathcal{E}^*$.)

The previous exercises suggest questions which are solved by the next results.

Proposition IX.3.16 (Soare [1974a]) *Every automorphism of $\mathcal{E}^*$ is induced by a permutation of ω.*

Proof. Let h be a presentation of the given automorphism. It is enough to find a permutation p such that, for every e,

$$R_e \ : \ p(\mathcal{W}_e) =^* \mathcal{W}_{h(e)}.$$

To obtain p one-one and onto, we use a back and forth argument like that in III.7.13. To satisfy R_e we want, except for at most finitely many x,

$$x \in \mathcal{W}_e \ \Leftrightarrow \ p(x) \in \mathcal{W}_{h(e)}.$$

Since we are allowed finitely many mistakes, at stage $n+1$ we only have to take care of the requirements R_e for $e \leq n$. Thus each e will be considered at almost every stage. To take care of all requirements at the same time we use e-states (as in the proof of III.4.18), and thus give priority to conditions with lower indices. Since we are not trying to make p recursive, there is no need to have an effective construction, and thus we can just consider:

$$\begin{aligned}\sigma(e,x) &= \sum\{2^{e-i} : i \leq e \ \wedge \ x \in \mathcal{W}_i\} \\ \tau(e,x) &= \sum\{2^{e-i} : i \leq e \ \wedge \ x \in \mathcal{W}_{h(i)}\}.\end{aligned}$$

We start with $p_0 = \emptyset$. At stage $n+1$:

- *if n is even, we work for totality*
  Let x_n be the smallest element not yet in the domain of p_n, and take the greatest $e \leq n$ such that there is y not yet in the range of p_n and such that $\sigma(e, x_n) = \tau(e, y)$, i.e.
  $$(\forall i \leq e)(x_n \in \mathcal{W}_i \ \Leftrightarrow \ y \in \mathcal{W}_{h(i)}).$$
  Let p_{n+1} be the extension of p_n that takes, as a value for x_n, the smallest such y if such an e exists, and the smallest element not in the range of p_n otherwise.

- *if n is odd, we work for ontoness*
  We interchange the two sides, i.e. take y_n not in the range of p_n and look for x.

By construction, $p = \bigcup_{n \in \omega} p_n$ is a permutation. Since Φ is an automorphism of $\boldsymbol{\mathcal{E}^*}$, we have

$$\begin{array}{rcl} \mathcal{W}_e \text{ (co)finite} & \Leftrightarrow & \mathcal{W}_{h(e)} \text{ (co)finite} \\ \Phi((\mathcal{W}_e \cap \mathcal{W}_i)^*) & = & (\mathcal{W}_{h(e)} \cap \mathcal{W}_{h(i)})^* \\ \Phi((\mathcal{W}_e \cup \mathcal{W}_i)^*) & = & (\mathcal{W}_{h(e)} \cup \mathcal{W}_{h(i)})^*. \end{array}$$

From this it easily follows that, given two e-states σ_0 and τ_0 such that $\sigma_0 = \tau_0$, there are infinitely many x in e-state σ_0 if and only if there are infinitely many y in e-state τ_0. The construction ensures that, if (x_n, y_n) is the pair defined at stage $n+1$,
$$(\exists n_0)(\forall n \geq n_0)[\sigma(e, x_n) = \tau(e, y_n)],$$
and so
$$(\forall i \leq e)(p(\mathcal{W}_i) =^* \mathcal{W}_{h(i)}). \quad \Box$$

Corollary IX.3.17 *Every automorphism of $\boldsymbol{\mathcal{E}^*}$ is induced by an automorphism of $\boldsymbol{\mathcal{E}}$.*

Proof. The permutation p defined above induces an automorphism of $\boldsymbol{\mathcal{E}}$ because $p(\mathcal{W}_e)$ is r.e., since it differs finitely from $\mathcal{W}_{h(e)}$. $\Box$

Theorem IX.3.18 (Lachlan) *There are $2^{\aleph_0}$ automorphisms of $\boldsymbol{\mathcal{E}^*}$.*

Proof. The idea is to split ω into infinite and uniformly recursive parts R_n, and to consider: on the one hand, nontrivial uniformly recursive permutations h_0^n of R_n; on the other hand, the identity functions h_1^n on R_n. Then we can build a permutation of ω by choosing, for each n, one of h_0^n and h_1^n, and gluing them together. In other words, any 0,1-valued function f defines a permutation

$$h_f = \bigcup_{n\in\omega} h^n_{f(n)}$$

of ω, and different functions define different permutations. By the given uniformities, each 0,1-valued recursive f produces a recursive h_f and hence an automorphism of $\boldsymbol{\mathcal{E}}^*$. This proves that there are at least countably many such automorphisms.

The argument does not produce $2^{\aleph_0}$ automorphisms because, in general, not every h_f induces an automorphism. Indeed, given an r.e. set A,

$$h_f(A) = \bigcup_{n\in\omega} h^n_{f(n)}(A \cap R_n)$$

need not be r.e., being an arbitrary union of r.e. sets $h^n_{f(n)}(A\cap R_n)$. Since finite unions of r.e. sets are r.e., we modify the previous approach to ensure that we only consider finitely many pieces, for every r.e. set.

We thus choose the infinite recursive sets R_n by splitting ω in such a way that, for every e,

$$\mathcal{W}_e \subseteq \bigcup_{n\leq e} R_n \quad \text{or} \quad \bigcup_{n>e} R_n \subseteq \mathcal{W}_e,$$

so that only finitely many of the $\mathcal{W}_e \cap R_n$ will be nontrivial. Since we do not require *uniform* recursiveness, this is easy to achieve by induction, as follows. Given e, see if $(\overline{\bigcup_{n<e} R_n}) \cap \mathcal{W}_e$ is finite. If so, we can choose R_e as an infinite recursive set covering it and such that $\bigcup_{n\leq e} R_n$ is still coinfinite, so that all the R_n with $n > e$ are going to be disjoint from $\mathcal{W}_e$. Otherwise, $(\overline{\bigcup_{n<e} R_n}) \cap \mathcal{W}_e$ is an infinite r.e. set, and it thus has an infinite recursive subset B: we can let R_e be the complement of $B \cup (\bigcup_{n<e} R_n)$, so that all the R_n with $n > e$ are now going to be subsets of B, and hence of $\mathcal{W}_e$.

We can now proceed as above, by choosing a nontrivial recursive permutation h_0^n of R_n, as well as the identity function h_1^n on it. For every 0,1-valued function f, h_f now induces an automorphism because

$$\begin{aligned} h_f(\mathcal{W}_e) &= \bigcup_{n\in\omega} h^n_{f(n)}(\mathcal{W}_e \cap R_n) \\ &= [\bigcup_{n\leq e} h^n_{f(n)}(\mathcal{W}_e \cap R_n)] \cup [\bigcup_{n>e} R_n \cap \mathcal{W}_e], \end{aligned}$$

and the expression on the right is now r.e. (because, by construction, the last part is either empty or equal to $\bigcup_{n>e} R_n = \overline{\bigcup_{n\leq e} R_n}$, and hence a recursive set). Since different f's induce different h_f's, there are $2^{\aleph_0}$ automorphisms. $\square$

Exercises IX.3.19 Recursively presentable automorphisms of $\mathcal{E}^*$.

a) *Every Δ_2^0-presentable automorphism of $\mathcal{E}^*$ is recursively presentable.* (Jockusch) (Hint: by the Limit Lemma IV.1.17.)

b) *Every recursive permutation induces a recursively presentable automorphism of $\mathcal{E}^*$.*

c) *Not every Δ_2^0 permutation induces an automorphism of $\mathcal{E}^*$.* (Hint: take a recursive, infinite and coinfinite set A, and define a permutation that sends A onto $\mathcal{K}$ and $\overline{A}$ onto $\overline{\mathcal{K}}$. This does not induce an automorphism, because it does not preserve recursive enumerability.)

d) *Not every recursively presentable automorphism of $\mathcal{E}^*$ is induced by a recursive permutation.* (Martin) (Hint: in the notation of the proof of IX.3.24, to which we refer, the stage s_e can be obtained recursively in $\mathcal{K}$. Thus there is a function f recursive in $\mathcal{K}$ such that $p(\mathcal{W}_e) = \mathcal{W}_{f(e)}$. Such an automorphism is then Δ_2^0-presentable and, by part a), recursively presentable. But it cannot be induced by a recursive permutation because, otherwise, it would preserve recursively invariant properties.)

e) *Every recursively presentable automorphism of $\mathcal{E}^*$ is induced by a Δ_3^0 permutation.* (Soare [1974a]) (Hint: by the proof of IX.3.16 applied to a recursive h.)

Cholak and Downey [1994] have proved that *not every recursively presentable automorphism of $\mathcal{E}^*$ is induced by a Δ_2^0 permutation.*

Exercise IX.3.20 Automorphism bases. *If $\mathcal{C}$ is a nontrivial class of r.e. sets closed under recursive isomorphisms, then the set $\mathcal{C}^*$ of equivalence classes of elements of $\mathcal{C}$ is an automorphism base for $\mathcal{E}^*$.* (Shore [1977]) (Hint: let Φ_1 and Φ_2 be automorphisms agreeing on $\mathcal{C}^*$ but not on $\mathcal{E}^*$. Since the nowhere simple sets generate $\mathcal{E}^*$ by IX.2.7.b, there is A nowhere simple such that $\Phi_1(A) \neq^* \Phi_2(A)$. Suppose $\Phi_1(A) - \Phi_2(A)$ is infinite (the other case being symmetric). $\Phi_2(A)$ is nowhere simple, since A is and Φ_2 preserves definable properties; then there is a recursive set $R \subseteq \Phi_1(A) - \Phi_2(A)$. Since $\mathcal{C}$ is nontrivial and closed under recursive isomorphisms, there is $C \in \mathcal{C}$ such that $\overline{C} \subseteq R$. Then $C \cup \Phi_1(A) = \omega$ and $C \cup \Phi_2(A) \neq \omega$, which is a contradiction.)

In particular, $\mathbf{REC}^*$ *is an automorphism base for $\mathcal{E}^*$.* On the other hand, Shore [1977] and Soare have proved that *not every automorphism of* $\mathbf{REC}^*$ *can be extended to an automorphism of $\mathcal{E}^*$*, and the same is true for many other natural classes of r.e. sets.

After having investigated automorphisms of $\mathcal{E}$ and $\mathcal{E}^*$ for their own sake, we now turn to their applications to properties of r.e. sets. We first prove that a result similar to IX.3.2 holds for properties invariant under automorphisms.

Proposition IX.3.21 (Soare [1974a]) *A property of r.e. sets well-defined on $\mathcal{E}^*$ (i.e. invariant under finite differences) is invariant under automorphisms of $\mathcal{E}^*$ if and only if it is invariant under automorphisms of $\mathcal{E}$.*

Proof. It is enough to show that any two infinite and coinfinite r.e. sets A and B are automorphic in $\mathcal{E}$ if and only if their equivalence classes A^* and B^* are automorphic in $\boldsymbol{\mathcal{E}}^*$.

If A and B are automorphic in $\boldsymbol{\mathcal{E}}$ then, since every automorphism of $\boldsymbol{\mathcal{E}}$ induces one of $\boldsymbol{\mathcal{E}}^*$, A^* and B^* are automorphic in $\boldsymbol{\mathcal{E}}^*$.

Conversely, if A^* and B^* are automorphic in $\boldsymbol{\mathcal{E}}^*$, then one can start the proof of IX.3.16 with an enumeration of the r.e. sets such that $\mathcal{W}_0 = A$ and $\mathcal{W}_{h(0)} = B$, so that $p(A) = B$ by construction. One thus obtains an automorphism of $\boldsymbol{\mathcal{E}}$ interchanging A and B. $\square$

Exercises IX.3.22 a) *A definable property is invariant under automorphisms.*

b) *A property invariant under automorphisms is recursively invariant.* (Hint: recursive permutations induce automorphisms.)

c) *There are recursively invariant properties that are not invariant under automorphisms.* (Rogers [1967]) (Hint: let A be a simple set, $z \in \overline{A}$ and $B = A \cup \{z\}$. Then, as in VI.5.1, A and B are not recursively isomorphic. To obtain an automorphism sending A to B, let $C \subseteq A$ and $D \subseteq \overline{A} - \{z\}$ be cohesive sets, and if $C = \{c_0, c_1, \ldots\}$ and $D = \{d_0, d_1, \ldots\}$, let

$$p(z) = c_0 \qquad p(c_n) = c_{n+1} \qquad p(d_{n+1}) = d_n \qquad p(d_0) = z,$$

and $p(x) = x$ if $x \notin C \cup D \cup \{z\}$. Then p induces an automorphism, as in IX.3.14.a.)

d) *There are properties invariant under automorphisms that are not definable.* (Soare) (Hint: if A is the intersection of n maximal sets differing infinitely, then $\mathcal{L}^*(A)$ is the Boolean algebra of 2^n elements. Thus, for each set X, the property of being, for some $n \in X$, the intersection of n maximal sets differing infinitely is invariant under automorphisms. This determines $2^{\aleph_0}$ such properties, but only countably many can be definable.)

Harrington and Nies [1998] and Nies [199?a] have proved that *there are properties of r.e. sets invariant under automorphisms of $\boldsymbol{\mathcal{E}}^*$ and definable in First-Order Arithmetic, but not definable in $\boldsymbol{\mathcal{E}}^*$.*

We now turn to properties that are not invariant under automorphisms, and hence not definable.

Exercise IX.3.23 *The automorphisms constructed in IX.3.18 cannot be used to prove that there are properties that are recursively invariant, but not invariant under automorphisms.* (Hint: see the last part of the proof of IX.3.18.)

Theorem IX.3.24 (Martin) *The property of being hypersimple is not invariant under automorphisms of $\boldsymbol{\mathcal{E}}$.*

Proof. We want to define a hypersimple set A and a permutation p inducing an automorphism of $\boldsymbol{\mathcal{E}}$, such that $p(A)$ is not hypersimple. Obviously p cannot be a recursive permutation, otherwise it would preserve recursively invariant properties (including hypersimplicity). We will obtain p as the limit of recursive permutations p_s.

To make $p(A)$ not hypersimple, by III.3.8.3 it is enough to fix a recursive function f such that, for each n, $p(A)$ has at least n elements smaller than $f(n)$. For example, we can let $f(n) = 2n$ and ensure that at most $n+1$ of the elements up to $2n$ go into $p(A)$.

To make A hypersimple, as usual we will satisfy:

$$P_e \quad : \quad \mathcal{W}_e \text{ infinite disjoint strong array } \Rightarrow (\exists z \in \mathcal{W}_e)(D_z \subseteq A).$$

When D_z goes into A_{s+1} we have no control over $p_s(D_z)$, and too many elements may go into $p(A)$ below $2n$. Suppose e.g. that there are already $n+1$, and $p_s(x) \leq 2n$ for some $x \in D_z$. This would violate the restriction on $p(A)$, but we can modify the permutation p_s by interchanging x and y, where y is any element such that $p_s(y) > 2n$ and $y \notin A_s$. We thus define

$$p_{s+1}(a) = \begin{cases} p_s(y) & \text{if } a = x \\ p_s(x) & \text{if } a = y \\ p_s(a) & \text{otherwise.} \end{cases}$$

To avoid reference to unspecified n's, we just have to change the condition $p_s(y) > 2n$ into $p_s(y) \geq 2x$. Then for each $p_s(y)$ entering $p(A)$ there must be an x entering A such that $p_s(y) \geq 2x$, and if $2n \geq p_s(y) \geq 2x$, then $n \geq x$, i.e. this can happen at most $n+1$ times.

To ensure that p_s reaches a limit, notice that if

$$p_{s+1}(a) \neq p_s(a),$$

then a is either x or y as above, in particular none of them is in A_s. If a is x, then it enters A_{s+1} and is never considered again, so that from that point on $p_{s+1}(x)$ is final. If a is y, then there is no reason why it should not be considered again, for any number of other x's, and this may cause trouble. But we can just impose the condition $y > x$; then, once all the smaller elements that ever enter A have entered it, y is never considered again, and from that stage t on $p_t(y)$ is final.

To ensure that p_s^{-1} reaches a limit, we proceed similarly. If

$$p_{s+1}^{-1}(b) \neq p_s^{-1}(b),$$

then b is either $p_s(x)$ or $p_s(y)$ as above, and this happens for the satisfaction of some P_e. Now each condition requires only finite action (to put a finite set into A), but there are infinitely many conditions. Again we can just impose the restriction $e \leq b$, i.e. $e \leq p_s(x), p_s(y)$.

Since $p = \lim_{s\to\infty} p_s$ exists, it is a permutation because the p_s are, i.e. $p_s^{-1}(p_s(a)) = a$ and $p_s(p_s^{-1}(b)) = b$.

By construction $p(A)$ is coinfinite (since we leave out of it n elements below $2n$, for every n). Then so is A, because p^{-1} is a permutation.

We still have to ensure that p induces an automorphism of $\boldsymbol{\mathcal{E}}$. For this it is enough to interchange x and y at stage $s+1$ (for the sake of condition P_e) only if both they and their images via p_s are exactly in the same r.e. sets with indices $i \leq e$, i.e.

$$\begin{aligned} x \in \mathcal{W}_{i,s} \quad &\Leftrightarrow \quad y \in \mathcal{W}_{i,s} \\ p_s(x) \in \mathcal{W}_{i,s} \quad &\Leftrightarrow \quad p_s(y) \in \mathcal{W}_{i,s}. \end{aligned}$$

Since each condition requires attention only finitely often, from a certain stage s_e on only conditions with index greater than e will be considered, and so only elements with the same e-states are interchanged. Thus

$$p(\mathcal{W}_e) = \bigcup_{s \geq s_e} p_s(\mathcal{W}_{e,s}) \quad \text{and} \quad p^{-1}(\mathcal{W}_e) = \bigcup_{s \geq s_e} p_s^{-1}(\mathcal{W}_{e,s}),$$

and hence both $p(\mathcal{W}_e)$ and $p^{-1}(\mathcal{W}_e)$ are r.e.

Putting together all the conditions found above, the construction is the following. Start with $A_0 = \emptyset$ and $p_0 =$ the identity function. At stage $s+1$, look for the smallest $e \leq s$ such that P_e has not yet been satisfied (i.e. the elements of $\mathcal{W}_{e,s}$ code finite disjoint sets, none of which is contained in A_s), and a step towards its satisfaction can be done now, i.e. for some $z \in \mathcal{W}_{e,s}$, $x \in D_z \cap \overline{A}_s$ and $y \in \overline{A}_s$:

- $p_s(y) \geq 2x$
- $y > x$
- $e \leq p_s(x), p_s(y)$
- for every $i \leq e$, x and y, as well as $p(x)$ and $p(y)$, are exactly in the same r.e. sets with indices $i \leq e$.

If such an e does not exist, do nothing (i.e. $A_{s+1} = A_s$ and $p_{s+1} = p_s$). Otherwise, take the smallest e, the smallest z for such an e, and the smallest x and y for such a z: add x to A, and interchange x and y as explained above.

By the discussion already given the limit p of the p_s exists, it is a permutation, and it induces an automorphism of $\boldsymbol{\mathcal{E}}$. Moreover, A is hypersimple but $p(A)$ is not. $\square$

Exercise IX.3.25 *In general, Turing degrees are not invariant under automorphisms of* $\boldsymbol{\mathcal{E}}$. (Martin) (Hint: use the method of IX.3.24 to build A r.e. and p limit of recursive permutations, such that $A \not\equiv_T p(A)$.)

As a further example of negative result, Stob [1982a] has proved that *dense simplicity is not invariant under automorphisms.*

Orbits $\star$

A notion of equivalence similar to that of recursive isomorphism type can be defined as follows:

$$A \equiv_{\mathcal{E}} B \Leftrightarrow \text{there is an automorphism of } \boldsymbol{\mathcal{E}} \text{ which sends } A \text{ into } B.$$

Equivalence classes are called **$\boldsymbol{\mathcal{E}}$-orbits**. Each orbit consists of recursive isomorphism types (by IX.3.22.a), and all sets in it satisfy the same properties definable in $\boldsymbol{\mathcal{E}}$ (by IX.3.22.a).

Examples of orbits are the following:

- the *infinite, coinfinite recursive sets* (all infinite, coinfinite recursive sets are recursively isomorphic (p. I.327), hence automorphic by IX.3.14.c; conversely, by IX.3.3 the notion of infinite, coinfinite recursive set is definable, and hence preserved under automorphisms)
- the *creative sets* (similarly, by III.7.14 and IX.3.5)
- the *maximal sets* (Soare [1974a]; see Cholak [1994] for a different proof)
- the *hemimaximal sets*, defined as halves of r.e. splittings of maximal sets (Downey and Stob [1992]).

While the first two examples consist of only one recursive isomorphism type, the last two contain sets of different degrees.

Various *connections between orbits and degree structure* are known, in particular:

- there is a noncreative orbit consisting only of T-complete sets (Harrington and Soare [199?])
- some orbits contain a T-complete set, e.g. the orbits containing a low simple set (Downey and Stob [1992]), or a set of promptly simple degree (Cholak, Downey and Stob [1992], Harrington and Soare [199?b]), but not every nonrecursive orbit does (Harrington and Soare [1991])
- every nonrecursive orbit contains a set of a high degree (Cholak [1995], Harrington and Soare [199?b])
- some orbit contains sets of every high r.e. degree, e.g. the orbit of maximal sets (Soare [1974]), but not every nonrecursive orbit does (Maass, Shore and Stob [1981], see X.6.9.d)
- some orbit contains sets realizing all possible jumps of r.e. degrees, e.g. the orbit of hemimaximal sets (Downey and Stob [1992]), but no orbit contains sets of every nonzero r.e. degree (Harrington).

As for the *connection between orbits and lattice structure*, the following implication is trivial:

$$A \equiv_{\mathcal{E}} B \;\Rightarrow\; \mathcal{L}^*(A) \cong \mathcal{L}^*(A).$$

In general the converse implication fails, even when $\mathcal{L}(A) \cong \boldsymbol{\mathcal{E}}$ (Lerman and Soare [1980a], Soare [1982]), or when $\mathcal{L}^*(A)$ is a Boolean algebra (Lerman, Shore and Soare [1978], Herrmann [1983]). However, the converse implication does hold in some special cases, e.g. for Boolean algebras isomorphic via a Σ^0_3-definable isomorphism (Maass [1984]; see Cholak [1994] for a different proof), in particular for finite Boolean algebras with the same number of elements (Soare [1974]).

Cholak, Downey and Harrington [199?] have proved that *the index set* $\{\langle e, i\rangle : \mathcal{W}_e \equiv_{\mathcal{E}} \mathcal{W}_i\}$ *is* Σ^1_1*-complete*. This implies that, for any n, there are r.e. sets A and B that are automorphic, but not via Δ^0_n-presentable automorphisms, and similarly for all levels of the Hyperarithmetical Hierarchy (see p. I.391).

Definable and invariant classes of r.e. degrees ⋆

The following natural notions transfer lattice-theoretical properties to degrees.

Definition IX.3.26 *A class* $\boldsymbol{\mathcal{C}}$ *of r.e. degrees is* **(lattice-theoretically) definable** *if there is a property* P *of r.e. sets definable in* $\boldsymbol{\mathcal{E}}^*$ *such that a degree is in* $\boldsymbol{\mathcal{C}}$ *if and only if it contains an r.e. set satisfying* P*.*

Similarly, a class $\boldsymbol{\mathcal{C}}$ *of r.e. degrees is* **(lattice-theoretically) invariant** *if there is a property* P *of r.e. sets invariant under automorphisms of* $\boldsymbol{\mathcal{E}}^*$ *such that a degree is in* $\boldsymbol{\mathcal{C}}$ *if and only if it contains an r.e. set satisfying* P*.*

Since any definable property is invariant under automorphisms, existence results are stronger when stated in terms of definable classes, while nonexistence results are stronger when stated in terms of invariant classes.

On the positive side, the following jump classes (see XI.1.9) are definable, by the following properties:

$\mathbf{L}_0 = \{\mathbf{0}\}$	recursive sets
$\overline{\mathbf{L}}_0 = \{\boldsymbol{a} : \boldsymbol{a} > \mathbf{0}\}$	r.e. nonrecursive sets; simple sets (III.2.14)
$\overline{\mathbf{L}}_2 = \{\boldsymbol{a} : \boldsymbol{a}'' > \mathbf{0}''\}$	r.e. sets without maximal supersets (IX.2.29
$\mathbf{H}_1 = \{\boldsymbol{a} : \boldsymbol{a}' = \mathbf{0}''\}$	maximal sets (IX.2.24), r-maximal sets (IX.2.25.c)
	hyperhypersimple sets (IX.2.25.b)
$\mathbf{H}_0 = \{\mathbf{0}'\}$	creative sets (IX.3.5).

Moreover, $\mathbf{H}_n$ and $\overline{\mathbf{L}}_n$ are invariant, for all $n \geq 2$ (Cholak and Harrington [199?]).

On the negative side, the following classes are not invariant: $\mathbf{L}_{n+1}$ and $\overline{\mathbf{H}}_{n+1}$ (Cholak [1995], and Harrington and Soare [199?b]), and $\overline{\mathbf{L}}_1$ (Harrington and Soare [199?a]).

In other words, among the jump classes and their complements, the invariant ones are exactly $\mathbf{L}_0$ and

$$\overline{\mathbf{L}}_0 \supset \overline{\mathbf{L}}_2 \supset \overline{\mathbf{L}}_3 \supset \cdots \supset \mathbf{H}_2 \supset \mathbf{H}_1 \supset \mathbf{H}_0.$$

Lerman and Soare [1980a] have proved that *not every definable class is expressible in terms of jump classes*, since there is a definable class which nontrivially splits $\mathbf{L}_1$. Maass, Shore and Stob [1981] have improved on this, by showing that *there is a definable class that nontrivially splits every* $\mathbf{L}_n$ *and* $\mathbf{H}_n$*, for* $n > 0$ (see X.6.9.d).

In a different direction, Downey and Stob [1992] have proved that *not every nontrivial definable class is upward closed*, although all examples above are.

IX.4 Complexity of R.E. Sets

In Section VII.5 we developed a complexity theory for *total* recursive functions. In particular, we noticed on p. 37 that every total recursive function has a best complexity modulo some recursive function, for a trivial reason: if e is an index for the given function, its complexity Φ_e is total, and best modulo itself. It is only in the context of *classes* of functions that the notion of best complexity becomes interesting, and the Speed-Up Theorem VII.2.16 shows that there is no fixed modulus that works for every recursive function.

Similarly, and again trivially, every partial recursive function has a best complexity modulo some *partial* recursive function with the same domain: if e is an index for the given function, its complexity Φ_e has the same domain, and is best modulo itself. We now turn to the more intriguing notion of a partial recursive function with or without best complexity modulo *total* recursive functions.

Since when talking of speed of computations we are not interested in what the values of a function are, but only in how much resource they take to be computed, we only consider the complexity of r.e. sets, and briefly return to partial recursive functions in IX.4.2.

Nonspeedable sets

The crucial notion is the following.

Definition IX.4.1 (Blum and Marques [1973], Filotti) *A set A is* **nonspeedable** *if it is r.e. and it has a best complexity modulo some recursive* *function g. Precisely, there is an index e of A such that, for every index i of A,*

$$(\forall_\infty x)[\Phi_e(x) \le g(x, \Phi_i(x))].$$

A set A is **speedable** *if it is r.e. and not nonspeedable, i.e. it does not have a best complexity modulo any recursive function. Precisely, for every recursive function g and every index e of A there is another index i of A such that*

$$(\exists_\infty x)[g(x, \Phi_i(x)) < \Phi_e(x)].$$

Notice that the infinitely many x's such that

$$g(x, \Phi_i(x)) < \Phi_e(x)$$

must be elements of A, since if $x \notin A$, then neither $\Phi_i(x)$ nor $\Phi_e(x)$ converges, and hence both sides are undefined (and equal by convention).

Exercises IX.4.2 (Non)speedable partial recursive functions. A partial recursive function φ is **nonspeedable** if there is a recursive function g and an index e of φ such that, for every index i of φ,

$$(\forall_\infty x)[\Phi_e(x) \le g(x, \Phi_i(x))].$$

And φ is **speedable** if for every recursive function g and every index e of φ there is another index i of φ such that

$$(\exists_\infty x)[g(x, \Phi_i(x)) < \Phi_e(x)].$$

a) *If the domain of φ is nonspeedable as a set, then φ is nonspeedable as a function.* (Hint: if $\varphi \simeq \varphi_e$ then $\mathcal{W}_e$ is the domain of φ.)

b) *If φ is potentially recursive* (i.e. it has a total recursive extension, see II.2.2) *and φ is nonspeedable as a function, then its domain is nonspeedable as a set.* (Hint: let f be a total recursive extension of φ, and e be a program for φ with best complexity modulo g. Let

$$\varphi_{h(i)}(x) \simeq \begin{cases} f(x) & \text{if } x \in \mathcal{W}_i \\ \text{undefined} & \text{otherwise.} \end{cases}$$

If $\mathcal{W}_i$ is the domain of φ, then $\varphi_{h(i)} \simeq \varphi$ and thus, for almost every x,

$$\Phi_e(x) \le g(x, \Phi_{h(i)}(x)).$$

By maximization, there is a recursive function bounding $\Phi_{h(i)}$ in terms of Φ_i. If g is monotone (which can always be assumed), by composition there is a recursive function bounding Φ_e in terms of Φ_i.)

c) *If A is speedable as a set, then φ defined as*

$$\varphi(x) \simeq 0 \Leftrightarrow x \in A$$

is speedable as a function. (Hint: from part b).)

We now characterize the (non)speedable sets in recursion-theoretical terms. First we introduce the appropriate concepts, which are variations of the notion of jump.

Exercises IX.4.3 Relativized Halting Problem. Let

$$H^A = \{x : \mathcal{W}_x^A \neq \emptyset\}.$$

a) $H^A \equiv_1 A'$. (Selman [1974]) (Hint: H^A is r.e. in A, and so $H^A \leq_1 A'$. For the converse,

$$x \in A' \Leftrightarrow \mathcal{W}_{f(x)}^A \neq \emptyset,$$

where

$$\mathcal{W}_{f(x)}^A = \begin{cases} \omega & \text{if } x \in A' \\ \emptyset & \text{otherwise.)} \end{cases}$$

b) *A is recursive if and only if $H^A \equiv_1 \mathcal{K}$.* (Hint: if $H^A \leq_1 \mathcal{K}$, then $A \leq_1 H^A \leq_1 \mathcal{K}$ and A is r.e. Moreover, $H^A \equiv_1 H^{\overline{A}}$ and $\overline{A}$ is also r.e.)

c) *A is low if and only if $H^A \equiv_T \mathcal{K}$.*

d) *If an r.e. set A admits a recursive enumeration $\{A_s\}_{s\in\omega}$ such that, for all e,*

$$(\exists_\infty s)(\mathcal{W}_{e,s}^{A_s} \neq \emptyset) \Rightarrow \mathcal{W}_e^A \neq \emptyset,$$

then A is low. (Hint: let

$$g(e,s) = \begin{cases} 1 & \text{if } \mathcal{W}_{e,s}^{A_s} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

By the Limit Lemma, $H^A \leq_T \mathcal{K}$.)

See X.3.3 for another property in the style of part d), also ensuring lowness.

From IX.4.3.a (see also X.9.5), we notice that the set

$$H = \{x : \mathcal{W}_x \neq \emptyset\}$$

(called **Nem** in X.9.5) is recursively isomorphic to $\emptyset'$. The set H^A of the exercises above is its recursion-theoretical relativization, and hence an equivalent definition of the jump operator. We now consider a set-theoretical relativization.

Definition IX.4.4 (Hay [1973b], Selman [1974]) *The set*

$$H_A = \{x : \mathcal{W}_x \cap A \neq \emptyset\}$$

is called the **weak jump of *A***.

Recall that the notion of Q-reducibility was defined (in III.4.1) as follows: $A \leq_Q B$ if, for some recursive function f,

$$x \in A \Leftrightarrow \mathcal{W}_{f(x)} \subseteq B$$

or, equivalently,

$$x \in \overline{A} \Leftrightarrow \mathcal{W}_{f(x)} \cap \overline{B} \neq \emptyset.$$

The relationship with the notion of weak jump is evident, and its effects will surface repeatedly throughout this section.

Exercises IX.4.5 a) *$A \leq_Q B$ if and only if $H_{\overline{A}} \leq_1 H_{\overline{B}}$.* (Hint: if

$$x \in A \Leftrightarrow \mathcal{W}_{f(x)} \subseteq B,$$

then

$$\mathcal{W}_x \cap \overline{A} \neq \emptyset \Leftrightarrow (\bigcup_{z \in \mathcal{W}_x} \mathcal{W}_{f(z)}) \cap \overline{B} \neq \emptyset.$$

Conversely, if $H_{\overline{A}} \leq_1 H_{\overline{B}}$ then $\overline{A} \leq_1 H_{\overline{A}} \leq_1 H_{\overline{B}}$, so

$$x \in \overline{A} \Leftrightarrow \mathcal{W}_{f(x)} \cap \overline{B} \neq \emptyset$$

for some recursive f, and $A \leq_Q B$.)

b) *If $A \neq \emptyset$, then $A \oplus \mathcal{K} \leq_1 H_A \leq_1 A'$.* This justifies the name 'weak jump' given to H_A. (Hint: if

$$\mathcal{W}^A_{h(x)} = \begin{cases} \omega & \text{if } \mathcal{W}_x \cap A \neq \emptyset \\ \emptyset & \text{otherwise,} \end{cases}$$

then

$$x \in H_A \Leftrightarrow \mathcal{W}^A_{h(x)} \neq \emptyset,$$

and $H_A \leq_1 A'$ by IX.4.3.a.)

c) *If $A \neq \omega$ is r.e., then $\mathcal{K} \leq_1 H_{\overline{A}} \leq_1 \mathcal{K}'$.* (Hint: if A is r.e., then $H_{\overline{A}} \in \Sigma^0_2$, and $H_{\overline{A}} \leq_1 \mathcal{K}'$.)

d) *If $A \neq \omega$ is r.e., then A is recursive if and only if $H_{\overline{A}} \equiv_1 \mathcal{K}$.* (Hint: since $\overline{A} \leq_1 H_{\overline{A}}$, if $H_{\overline{A}} \leq_1 \mathcal{K}$, then $\overline{A}$ is r.e. and A is recursive. Conversely, if A is recursive, let $\mathcal{W}_{f(x)} = \mathcal{W}_x \cap \overline{A}$; then

$$x \in H_{\overline{A}} \Leftrightarrow \mathcal{W}_{f(x)} \neq \emptyset,$$

hence $H_{\overline{A}}$ is r.e. and $H_{\overline{A}} \leq_1 \mathcal{K}$.)

e) *If A is r.e., then A is Q-complete if and only if $H_{\overline{A}} \equiv_1 \mathcal{K}'$.* (Hint: by part a), $\mathcal{K} \leq_Q A$ is equivalent to $H_{\overline{\mathcal{K}}} \leq_1 H_{\overline{A}}$. It is enough to show that $\mathcal{K}' \equiv_1 H_{\overline{\mathcal{K}}}$, and this follows from X.9.10 and the fact that $H_{\overline{\mathcal{K}}}$ is Σ^0_2-complete.)

For more on the weak jump see X.9.10 and Hay [1973b], [1973c], Selman [1974], Morris [1976].

Recall that $A \oplus \mathcal{K} \leq_1 A'$, and a set is low if it has the least possible jump $\mathcal{K}$. Since $A \oplus \mathcal{K} \leq_1 H_A$ (for $A \neq \emptyset$), the least possible weak jump is again $\mathcal{K}$.

Definition IX.4.6 *A set A is* **semilow** *if $H_A \equiv_T \mathcal{K}$ (or, equivalently for $A \neq \emptyset$, $H_A \leq_T \mathcal{K}$).*

Exercises IX.4.7 a) *Low $\Rightarrow$ semilow, but the converse fails even for co-r.e. sets.* (Hint: $H_A \leq_1 A'$. The counterexample will follow from IX.4.14, which shows that every r.e. T-degree contains a semilow co-r.e. set.)

b) *$A \neq \emptyset$ is semilow if and only if there is a 0,1-valued recursive function g such that*

$$\begin{array}{lcl} \mathcal{W}_x \cap A \neq \emptyset & \Rightarrow & \lim_{s\to\infty} g(x,s) = 1 \\ \mathcal{W}_x \cap A = \emptyset & \Rightarrow & \lim_{s\to\infty} g(x,s) = 0. \end{array}$$

(Hint: by the Limit Lemma IV.1.17.)

c) *If an r.e. set A admits a recursive enumeration $\{A_s\}_{s\in\omega}$ such that, for all e,*

$$(\exists_\infty s)(\mathcal{W}_{e,s} \cap \overline{A}_s \neq \emptyset) \ \Rightarrow\ \mathcal{W}_e \cap \overline{A} \neq \emptyset,$$

then $\overline{A}$ is semilow. (Hint: similar to IX.4.3.d.)

Theorem IX.4.8 Recursion-Theoretic Characterization of the Nonspeedable Sets (Soare [1977]) *An r.e. set A is nonspeedable if and only if $\overline{A}$ is semilow.*

Proof. Suppose A is nonspeedable: for some recursive function g and some index e of A,

$$A = \mathcal{W}_i \ \Rightarrow\ (\forall_\infty x)[\Phi_e(x) \leq g(x, \Phi_i(x))].$$

We want to prove that $H_{\overline{A}} \leq_T \mathcal{K}$. Since A is r.e., $H_{\overline{A}}$ is automatically Σ^0_2, and thus it suffices to show that $H_{\overline{A}}$ is also Π^0_2, i.e. that $\overline{H}_{\overline{A}} \in \Sigma^0_2$. This follows from the following representation, whose matrix is Π^0_1 and where h is a recursive function such that $\mathcal{W}_{h(i)} = A \cup \mathcal{W}_i$:

$$\mathcal{W}_i \cap \overline{A} = \emptyset \ \Leftrightarrow\ (\exists n)(\forall x)[\Phi_e(x) \leq g(x, \Phi_{h(i)}(x)) + n].$$

Indeed:

- If $\mathcal{W}_i \cap \overline{A} = \emptyset$, then $\mathcal{W}_{h(i)} = A$ and so

  $$(\forall_\infty x)[\Phi_e(x) \leq g(x, \Phi_{h(i)}(x))].$$

  If x_0 is such that $\Phi_e(x) \leq g(x, \Phi_{h(i)}(x))$ for all $x \geq x_0$ and

  $$n = \max\{\Phi_e(x) : x < x_0 \wedge \Phi_e(x)\downarrow\},$$

  then $\Phi_e(x) \leq g(x, \Phi_{h(i)}(x)) + n$ for all x.

- If there is an n such that
$$(\forall x)[\Phi_e(x) \leq g(x, \Phi_{h(i)}(x)) + n],$$
then $\Phi_e(x)$ converges whenever $\Phi_{h(i)}(x)$ converges, and so
$$\mathcal{W}_i \subseteq \mathcal{W}_{h(i)} \subseteq \mathcal{W}_e = A,$$
hence $\mathcal{W}_i \cap \overline{A} = \emptyset$.

Suppose now $\overline{A}$ is semilow; by the Limit Lemma, there is a recursive 0,1-valued function f such that
$$\begin{array}{lcl} \mathcal{W}_x \cap \overline{A} \neq \emptyset & \Rightarrow & \lim_{s\to\infty} f(x,s) = 1 \\ \mathcal{W}_x \cap \overline{A} = \emptyset & \Rightarrow & \lim_{s\to\infty} f(x,s) = 0. \end{array}$$
Given e such that $\mathcal{W}_e = A$, we want to find a recursive function g such that
$$A = \mathcal{W}_i \ \Rightarrow\ (\forall_\infty x)[\Phi_e(x) \leq g(x, \Phi_i(x))].$$
The obvious guess, given i such that $\mathcal{W}_i = A$, is to let
$$h(i,x,z) = \begin{cases} \Phi_e(x) & \text{if } \Phi_i(x) \simeq z \\ 0 & \text{otherwise} \end{cases}$$
and
$$g(x,z) = \max\{h(i,x,z) : i \leq x\}.$$
Since, however, i is not always an index of A, h would not be total (for example, if $x \in \mathcal{W}_i \cap \overline{A}$ then $\Phi_i(x)$ converges but $\Phi_e(x)$ does not). But in the only troublesome case, namely when $\mathcal{W}_i \cap \overline{A} \neq \emptyset$, we have $f(i,s) = 1$ for almost every s. Thus we can let
$$s_x = \mu s \geq x.\,(\Phi_e(x) \leq s \ \vee\ f(i,s) = 1)$$
(notice that s_x is also a function of i).

When $\mathcal{W}_i \cap \overline{A} = \emptyset$, we have $f(i,s) = 1$ only for finitely many s, and thus for almost every x
$$s_x = \mu s \geq x.\,(\Phi_e(x) \leq s) = \max\{x, \Phi_e(x)\}.$$
Also, s_x is always defined when $x \in \mathcal{W}_i$. Indeed, either $x \in \overline{A}$ and thus $\mathcal{W}_i \cap \overline{A} \neq \emptyset$ and $f(i,s) = 1$ for almost every s, or $x \in A$ and $\Phi_e(x)\downarrow$. Thus we can let
$$h(i,x,z) = \begin{cases} \Phi_e(x) & \text{if } \Phi_i(x) \simeq z \ \wedge\ \Phi_e(x) \leq s_x \\ 0 & \text{otherwise} \end{cases}$$

and

$$g(x,z) = \max\{h(i,x,z) : i \le x\}.$$

If $\mathcal{W}_i = A$, then $\mathcal{W}_i \cap \overline{A} = \emptyset$ and $\Phi_e(x) \le s_x$ for almost every x, so that

$$\Phi_e(x) \le g(x, \Phi_i(x))$$

for almost every x. $\Box$

Corollary IX.4.9 *An r.e. set A is nonspeedable if and only if $H_{\overline{A}} \in \Pi^0_2$.*

Proof. As noticed in the proof above, if A is r.e., then $H_{\overline{A}} \in \Sigma^0_2$, and thus $H_{\overline{A}} \le_T \mathcal{K}$ if and only if $H_{\overline{A}} \in \Pi^0_2$. $\Box$

Notice that the second part of the proof above shows that *if A is nonspeedable, then every program for A has a best complexity modulo some recursive function* (Filotti).

Exercises IX.4.10 Other characterizations of nonspeedable sets.
a) *An r.e. set A is nonspeedable if and only if, for some recursive function f and every index e,*

$$\mathcal{W}_e \cap \overline{A} = \mathcal{W}_{f(e)} \cap \overline{A} \quad \text{and} \quad \mathcal{W}_e \subseteq A \Rightarrow \mathcal{W}_{f(e)} \textit{ finite.}$$

(Blum and Marques [1973], Morris [1974]) (Hint: if

$$\begin{array}{lcl} \mathcal{W}_e \cap \overline{A} \neq \emptyset & \Rightarrow & \lim_{s\to\infty} h(e,s) = 1 \\ \mathcal{W}_e \cap \overline{A} = \emptyset & \Rightarrow & \lim_{s\to\infty} h(e,s) = 0, \end{array}$$

let

$$\mathcal{W}_{f(e)} = \bigcup_{s\in\omega} \{x : x \in \mathcal{W}_{e,s} \wedge h(e,s) = 1\}.$$

Conversely, given f and an enumeration $\{A_s\}_{s\in\omega}$ of A, let

$$h(e,s) = \begin{cases} 1 & \text{if } \mathcal{W}_{f(e),s} \cap \overline{A}_s \neq \emptyset \\ 0 & \text{otherwise.)} \end{cases}$$

b) *An r.e. set A is nonspeedable if and only if, for some recursive function f and every index e,*

$$\mathcal{W}_e \cap \overline{A} \subseteq \mathcal{W}_{f(e)} \quad \text{and} \quad \mathcal{W}_e \cap \overline{A} \textit{ finite} \Rightarrow \mathcal{W}_{f(e)} \textit{ finite.}$$

(Soare [1977], Bennison) (Hint: first note that the condition $\mathcal{W}_e \cap \overline{A} \subseteq \mathcal{W}_{f(e)}$ may be replaced by $\mathcal{W}_e \cap \overline{A} = \mathcal{W}_{f(e)} \cap \overline{A}$ by substituting $\mathcal{W}_e \cap \mathcal{W}_{f(e)}$ for $\mathcal{W}_{f(e)}$. Then note that condition 2 in part a) may be replaced by condition 2 in part b), by defining $\mathcal{W}_{f(e)}$ as the union over $e \in \omega$ of the sets

$$\{x : x \in \mathcal{W}_{e,s} \wedge (\forall y \le x)[h(t(e,y), s) = 1]\},$$

where $\mathcal{W}_{t(e,y)} = \{z : z \in \mathcal{W}_e \ \wedge \ z \geq y\}$.)

c) *The recursive sets are the only sets A for which there exists a recursive function f such that*

$$\mathcal{W}_e \cap \overline{A} \textit{ finite } \Rightarrow \mathcal{W}_e \cap \overline{A} = \mathcal{W}_{f(e)}.$$

(Alton [1971]) (Hint: let $\mathcal{W}_{h(e)} = \{e\}$. Then $\overline{A} = \bigcup_{e \in \omega} \mathcal{W}_{f(h(e))}$.)

d) *For no coinfinite r.e. set A does a recursive function f exist such that*

$$\mathcal{W}_{f(e)} \textit{ finite } \quad \textit{and} \quad \mathcal{W}_e \cap \overline{A} \textit{ finite } \Rightarrow \mathcal{W}_e \cap \overline{A} \subseteq \mathcal{W}_{f(e)}.$$

(Alton [1971]) (Hint: let

$$\mathcal{W}_{g(e)} = A \cup \mathcal{W}_{f(e)} \cup \{\text{the first element of } \overline{A \cup \mathcal{W}_{f(e)}}\}.$$

By the Fixed-Point Theorem, there is e such that $\mathcal{W}_e = \mathcal{W}_{g(e)}$. Then $\mathcal{W}_e \cap \overline{A}$ is finite, but $\mathcal{W}_e \cap \overline{A} \not\subseteq \mathcal{W}_{f(e)}$.)

The notion of part b) is called **uniformity for finite extensions**, and was introduced by Alton [1971] as a possible extension of the class of recursive sets, since for A recursive there is f recursive such that $\mathcal{W}_e \cap \overline{A} = \mathcal{W}_{f(e)}$. Parts c) and d) show that other natural approaches to such an extension fail.

In many ways, *the nonspeedable sets resemble the recursive sets*. Specifically, in terms of:

- weak jump behavior ($H_{\overline{A}}$ is T-equivalent to $\mathcal{K}$ if and only if A is nonspeedable, and 1-equivalent to $\mathcal{K}$ if and only if A is recursive)
- uniformity for finite extensions (see IX.4.10)
- lattice-theoretical properties ($\mathcal{L}(A)$ is in both cases effectively isomorphic to $\mathcal{E}$, see Soare [1982]).

Effectively speedable sets

Notice that *the speedable sets are upward closed w.r.t. Q-reducibility* (among the r.e. sets): if $A \leq_Q B$, then $H_{\overline{A}} \leq_1 H_{\overline{B}}$ (IX.4.5.a); and if $H_{\overline{B}} \leq_T \mathcal{K}$, then $H_{\overline{A}} \leq_T \mathcal{K}$ (i.e. if B is nonspeedable then so is A).

In particular, from the existence of speedable sets (proved in the next subsection), it follows that *all Q-complete sets are speedable.* We now show that they are a special class of speedable sets.

Definition IX.4.11 (Blum and Marques [1973]) *A set A is* **effectively speedable** *if it is r.e. and there is a recursive function f such that, for every total recursive function φ_j and every index e of A, $f(e,j)$ is an another index of A such that*

$$(\exists_\infty x)[\varphi_j(x, \Phi_{f(e,j)}(x)) < \Phi_e(x)].$$

In other words, a set A is effectively speedable if, given g recursive and an index e of A, not only can we assert the existence of another index i of A with complexity g-faster than e infinitely often, but we can also produce i in an effective way from e and any index j of g.

As we can expect from a series of results in Chapter III (for example, III.2.18, III.3.15, III.4.22, III.6.2), r.e. sets satisfying recursion-theoretical properties in an effective way are complete. The next result confirms these expectations in the case of speedability.

Theorem IX.4.12 Recursion-Theoretic Characterization of the Effectively Speedable Sets (Blum and Marques [1973], Gill and Morris [1974]) *A set is effectively speedable if and only if it is Q-complete.*

Proof. We use the following characterization (III.6.20): a set is Q-complete if and only if it is subcreative, where a set A is subcreative if it is r.e. and, for some recursive function g,

$$\mathcal{W}_z \subseteq \overline{A} \ \Rightarrow\ A \subset \mathcal{W}_{g(z)} \subseteq \overline{\mathcal{W}}_z.$$

- *subcreative $\Rightarrow$ effectively speedable*
  Let A be subcreative. To obtain f as in IX.4.11, the idea is to let

  $$\mathcal{W}_{f(e,j)} = \mathcal{W}_e \cup \Big(\bigcup_{n\in\omega} \mathcal{W}_{h(n)}\Big)$$

  for an appropriate recursive function h, and to ensure that if φ_j is total and $\mathcal{W}_e = A$, then

  $$\emptyset \neq \mathcal{W}_{h(n)} \subseteq A$$

  (so that $\mathcal{W}_{f(e,j)} = A$), and

  $$x \in \mathcal{W}_{h(n)} \ \Rightarrow\ \Phi_e(x) > \varphi_j(x, \Phi_{f(e,j)}(x)) + n$$

  (so that if $x \in \mathcal{W}_{h(n)}$, then $\Phi_e(x) > n$, and thus in $\mathcal{W}_{f(e,j)}$ there are infinitely many elements x such that $\Phi_e(x) > \varphi_j(x, \Phi_{f(e,j)}(x))$).

  To ensure that, under the hypotheses on j and e, $\mathcal{W}_{h(n)} \neq \emptyset$, we use the fact that A is subcreative to note that

  $$\mathcal{W}_{h(n)} = \emptyset \ \Rightarrow\ \mathcal{W}_{h(n)} \subseteq \overline{A} \ \Rightarrow\ A \subset \mathcal{W}_{g(h(n))}.$$

  Thus in this case we know that there is $x \in \mathcal{W}_{g(h(n))} \cap \overline{A}$, and thus $\Phi_e(x) = \infty$ (since $x \notin \mathcal{W}_e = A$). However, we cannot conclude that

  $$\Phi_e(x) > \varphi_j(x, \Phi_{f(e,j)}(x)) + n,$$

because we do not know whether $x \in \mathcal{W}_{f(e,j)}$, the reason being that x is an element of $\mathcal{W}_{g(h(n))}$, but in the definition of $\mathcal{W}_{f(e,j)}$ we only used $\mathcal{W}_{h(n)}$. We thus modify the definition of f, and instead let

$$\mathcal{W}_{f(e,j)} = \mathcal{W}_e \cup (\bigcup_{n\in\omega} \mathcal{W}_{t(n)}),$$

where $\mathcal{W}_{t(n)} \subseteq \mathcal{W}_{g(h(n))}$, for h and t defined simultaneously by the following procedure:

– generate $\mathcal{W}_{g(h(n))}$
– when a new element x is generated in $\mathcal{W}_{g(h(n))}$, put x into $\mathcal{W}_{t(n)}$ (to ensure that $\Phi_{f(e,j)}(x)\downarrow$)
– see if $\Phi_e(x) > \varphi_j(x, \Phi_{f(e,j)}(x)) + n$
– if so, put x into $\mathcal{W}_{h(n)}$ and stop; otherwise, continue with a new element.

Note that the definition actually requires a *double use of the Fixed-Point Theorem*: h is used in its own definition both directly through $g \circ h$, and indirectly through f and hence t; and h is used in the definition of t through $g \circ h$.

Now, as above, if φ_j is total and $\mathcal{W}_e = A$ we have $\mathcal{W}_{h(n)} \neq \emptyset$, otherwise from $\mathcal{W}_{h(n)} = \emptyset$ we deduce $\mathcal{W}_{g(h(n))} \cap \overline{A} \neq \emptyset$ and so $\mathcal{W}_{h(n)} \neq \emptyset$, which is a contradiction.

Similarly, $\mathcal{W}_{h(n)} \subseteq A$ because otherwise $\mathcal{W}_{h(n)} \subseteq \overline{A}$ (since, by definition, $\mathcal{W}_{h(n)}$ is a singleton), and by subcreativity $\mathcal{W}_{g(h(n))} \subseteq \overline{\mathcal{W}}_{h(n)}$, contradicting the definition of $\mathcal{W}_{h(n)}$ (which makes $\mathcal{W}_{h(n)} \subseteq \mathcal{W}_{g(h(n))}$).

Also, $\mathcal{W}_{t(n)} \subseteq A$ because if $x \in \mathcal{W}_{t(n)} \cap \overline{A}$, then $\Phi_e(x) = \infty$ and $\Phi_{f(e,j)}(x)\downarrow$, hence

$$\Phi_e(x) > \varphi_j(x, \Phi_{f(e,j)}(x)) + n$$

automatically, and so $x \in \mathcal{W}_{h(n)} \subseteq A$ by definition. Thus

$$\mathcal{W}_e = A \Rightarrow \mathcal{W}_{f(e,j)} = A,$$

and A is effectively speedable.

- *effectively speedable* $\Rightarrow$ *subcreative*
Let A be effectively speedable via f. To obtain g recursive such that

$$\mathcal{W}_z \subseteq \overline{A} \Rightarrow A \subset \mathcal{W}_{g(z)} \subseteq \overline{\mathcal{W}}_z,$$

we fix an index e of A and let $\mathcal{W}_{g(z)} = A \cup \mathcal{W}_{f(e,j)}$ for an appropriate j (obtained uniformly from z). Given z, we thus have to define φ_j in such a way that, when $\mathcal{W}_z \subseteq \overline{A}$:

1. $\mathcal{W}_{f(e,j)} \cap \overline{A} \neq \emptyset$
2. $\mathcal{W}_{f(e,j)} \cap \overline{A} \subseteq \overline{\mathcal{W}}_z$.

If we make sure that, if $\Phi_{f(e,j)}(x)\uparrow$ or $\Phi_e(x)\downarrow$, then $\varphi_j(x,y)\downarrow$ (i.e., by contrapositive, that $\varphi_j(x,y)\uparrow$ if $x \in \mathcal{W}_{f(e,j)} \cap \overline{A}$), then we can satisfy 1 trivially, by ensuring that φ_j is not total. This can be ensured by diagonalization, using the fact that (since A is speedable via f) if φ_j is total, then there are infinitely many x such that $\Phi_e(x) > \varphi_j(x, \Phi_{f(e,j)}(x))$.

To satisfy 2 we want to avoid having elements in $\mathcal{W}_{f(e,j)} \cap \overline{A} \cap \mathcal{W}_z$, and we can obtain this by defining $\varphi_j(x,y)$ in such a way that, when $\Phi_e(x)\uparrow$ and $\Phi_z(x)\downarrow$, then φ_j is total (so that $\mathcal{W}_{f(e,j)} = A$).

We define $\varphi_j(x,y)$ as follows, by induction on a fixed ordering of pairs. If (x,y) is the first pair for which $\varphi_j(x,y)$ is still undefined, then:

- if $\Phi_{f(e,j)}(x) \not\simeq y$, let $\varphi_j(x,y) \simeq 0$
- if $\Phi_{f(e,j)}(x) \simeq y$, then:
  - if $\Phi_e(x)$ converges first, let $\varphi_j(x,y) \simeq \Phi_e(x)$
  - if $\Phi_z(x)$ converges first, let $\varphi_j(x,y) \simeq 0$, and run a computation of $\Phi_e(x)$. At each step of this computation, until $\Phi_e(x)$ converges, let φ_j be equal to 0 for the first pair for which it is still undefined.

The existence of j (which, as it stands, is used in the definition of φ_j itself) is ensured by the Fixed-Point Theorem.

Suppose now $\mathcal{W}_z \subseteq \overline{A}$. First we prove that φ_j is not total. Otherwise,

$$\mathcal{W}_{f(e,j)} = A \qquad \text{and} \qquad (\exists_\infty x)[\varphi_j(x, \Phi_{f(e,j)}(x)) < \Phi_e(x)].$$

But this is impossible, since then

$$\varphi_j(x, \Phi_{f(e,j)}(x)) < \Phi_e(x)$$

never holds. If $x \notin A$, because $\Phi_e(x)\uparrow$ and $\Phi_{f(e,j)}(x)\uparrow$, and so both sides are undefined (and equal). If $x \in A$, because then

$$\varphi_j(x, \Phi_{f(e,j)}(x)) = \Phi_e(x)$$

by definition of φ_j, since the last clause can never be used. Indeed, then $\Phi_z(x)\uparrow$, as follows:

$$\begin{aligned}\Phi_{f(e,j)}(x) \simeq y \quad &\Rightarrow \quad x \in \mathcal{W}_{f(e,j)} = A \\ &\Rightarrow \quad x \notin \mathcal{W}_z \text{ (because } \mathcal{W}_z \subseteq \overline{A}) \\ &\Rightarrow \quad \Phi_z(x)\uparrow .\end{aligned}$$

Since φ_j is not total, there are x and y such that $\varphi_j(x,y)\uparrow$. By definition it must be $\Phi_{f(e,j)}(x) \simeq y$ and $\Phi_e(x)\uparrow$, hence $x \in \mathcal{W}_{f(e,j)} \cap \overline{A}$, and 1 is proved.

To prove 2, suppose there is $x \in \mathcal{W}_{f(e,j)} \cap \overline{A} \cap \mathcal{W}_z$. Then $\Phi_{f(e,j)}(x) \simeq y$ for some y, and $\Phi_e(x)\uparrow$ but $\Phi_z(x)\downarrow$. Then φ_j is total by the last clause of its definition, contradicting what was proved above. □

Corollary IX.4.13 *An r.e. set A is effectively speedable if and only if $H_{\overline{A}}$ is Σ^0_2-complete.*

Proof. If A is r.e., then (by IX.4.5.e) A is Q-complete if and only if $H_{\overline{A}} \equiv_1 \mathcal{K}'$, and $\mathcal{K}'$ is Σ^0_2-complete. □

Thus the complements of effectively speedable sets have the highest possible weak jump, while the complements of nonspeedable sets have the lowest possible jump, and *nonspeedable and effectively speedable sets are at opposite ends of the spectrum of r.e. sets* defined by the behavior of $H_{\overline{A}}$.

Existence theorems

We consider existence conditions in terms of both degrees and structural properties.

Proposition IX.4.14 (Marques [1975a]) *Every r.e. T-degree contains an r.e. nonspeedable set.*

Proof. Let B be an r.e. set. We want an r.e. nonspeedable set A such that $A \equiv_T B$.

To ensure $A \equiv_T B$, we choose a recursive sequence $\{F_n\}_{n\in\omega}$ of disjoint, finite, nonempty sets such that $\bigcup_{n\in\omega} F_n = \omega$. We put at most one element of F_n into A, and this exactly when $n \in B$. Thus $B \leq_T A$ because

$$n \in B \;\Leftrightarrow\; F_n \cap A \neq \emptyset,$$

and $A \leq_T B$ because to see whether $x \in A$ we only have to search for the unique n such that $x \in F_n$, ask if $n \in B$ and, if so, run the construction to see which is the unique element of F_n that goes into A.

To ensure that A is nonspeedable, we use IX.4.7.c and generate A in such a way that

$$(\exists_\infty s)(\mathcal{W}_{e,s} \cap \overline{A}_s \neq \emptyset) \Rightarrow \mathcal{W}_e \cap \overline{A} \neq \emptyset.$$

It is natural to introduce the following element as a witness of the fact that $\mathcal{W}_{e,s} \cap \overline{A}_s \neq \emptyset$:

$$a_e^s = \begin{cases} \mu z.\,(z \in \mathcal{W}_{e,s} \cap \overline{A}_s) & \text{if } \mathcal{W}_{e,s} \cap \overline{A}_s \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

The construction of A is as follows. At stage $s+1$, if $n \in B_{s+1} - B_s$ is the unique element generated by B at this stage, put the element

$$x_n = \mu x.\,(x \in F_n - \{a_0^s, \ldots, a_n^s\})$$

in A_{s+1}. Such an element always exists if we choose F_n in such a way that $|F_n| = n + 2$.

Note that a_e^s can enter A only if it belongs to some F_n with $n < e$, and n is generated in B after stage s. Thus, if s_0 is such that all elements of B less than e have already been generated in B_{s_0}, a_e^s cannot enter A at any stage $s \geq s_0$, and it can change only if it gets smaller. Hence $\lim_{s\to\infty} a_e^s = a_e$ exists, and

$$(\exists_\infty s)(\mathcal{W}_{e,s} \cap \overline{A}_s \neq \emptyset) \Rightarrow a_e \in \mathcal{W}_e \cap \overline{A}.$$

Thus $\overline{A}$ is semilow, and A is nonspeedable by IX.4.8. $\square$

Notice that the proof actually produces a *tt*-reduction of B to A, and a *wtt*-reduction (with bound n) from A to B. In particular, *every r.e. wtt-degree contains an r.e. nonspeedable set.*

Proposition IX.4.15 (Marques [1975a]) *Every low r.e. set is nonspeedable.*

Proof. This is a corollary of IX.4.8, since if A is low then $\overline{A}$ is low, and hence semilow. $\square$

The last result provides a necessary condition for an r.e. degree not to contain an r.e. speedable set. This also turns out to be sufficient.

Proposition IX.4.16 (Marques [1975a], Soare [1977]) *An r.e. T-degree contains an r.e. speedable set if and only if it is not low.*

Proof. By IX.4.15, it is enough to prove that if an r.e. degree $\boldsymbol{a}$ is not low, i.e. $\boldsymbol{a}' > \boldsymbol{0}'$, then it contains a speedable set.

Let $A \in \boldsymbol{a}$ be an r.e. set, and define

$$u \in B \Leftrightarrow D_u \cap A \neq \emptyset.$$

Then B is also r.e., and $B \equiv_T A$. To show that B is speedable, it is enough to show that $A' \leq_T H_{\overline{B}}$, since if $H_{\overline{B}} \leq_T \mathcal{K}$ (i.e. if B is not speedable), then $A' \leq_T \mathcal{K}$ (i.e. A is low), contradicting the hypothesis.

That $A' \leq_T H_{\overline{B}}$ is immediate. Since A' is r.e. in A, by III.1.4 there is a recursive function f such that

$$\begin{aligned} x \in A' &\Leftrightarrow (\exists u)(u \in \mathcal{W}_{f(x)} \wedge D_u \subseteq \overline{A}) \\ &\Leftrightarrow (\exists u)(u \in \mathcal{W}_{f(x)} \wedge u \in \overline{B}) \\ &\Leftrightarrow \mathcal{W}_{f(x)} \cap \overline{B} \neq \emptyset \\ &\Leftrightarrow f(x) \in H_{\overline{B}}. \quad \Box \end{aligned}$$

Proposition IX.4.17 (Jockusch [1969a], Blum and Marques [1973], Marques [1975], Soare [1977]) *Every finitely strongly hypersimple set is speedable.*

Proof. Suppose A is nonspeedable. Then, in particular, $\overline{A}$ is infinite and $H_{\overline{A}} \leq_T \mathcal{K}$ (we do not use the hypothesis that A is r.e. in the proof).

By definition III.4.9, to prove that A is not finitely strongly hypersimple we need a weak array of disjoint finite sets intersecting $\overline{A}$, and with union covering it.

Consider all possible weak arrays $\{\mathcal{W}_{\varphi_e(x)}\}_{x\in\omega}$, where $\mathcal{W}_{\varphi_e(x)} = \emptyset$ if $\varphi_e(x)\uparrow$. Since $H_{\overline{A}} \leq_T \mathcal{K}$, there is a 0,1-valued recursive function g such that

$$\begin{aligned} \mathcal{W}_{\varphi_e(x)} \cap \overline{A} \neq \emptyset &\Rightarrow \lim_{s\to\infty} g(e,x,s) = 1 \\ \mathcal{W}_{\varphi_e(x)} \cap \overline{A} = \emptyset &\Rightarrow \lim_{s\to\infty} g(e,x,s) = 0. \end{aligned}$$

For each e, we define a disjoint weak array $\{\mathcal{W}_{\varphi_{h(e)}(x)}\}_{x\in\omega}$ covering ω, as follows. At stage s we put s into one of the elements of the array. Consider the x's such that:

- $x \leq s$
- $g(e,x,s) = 0$.

If there is such an x, choose the least one x_s, and let

$$s \in \mathcal{W}_{\varphi_{h(e)}(x_s)}.$$

If there is no such x, let

$$s \in \mathcal{W}_{\varphi_{h(e)}(s)}.$$

If x is the smallest integer such that $\mathcal{W}_{\varphi_e(x)} \cap \overline{A} = \emptyset$, then $g(e,x,s) = 0$ from a certain point on, and as x is the smallest one for which this happens, from a certain point on all the elements go into $\mathcal{W}_{\varphi_{h(e)}(x)}$, hence $\mathcal{W}_{\varphi_{h(e)}(x)} \cap \overline{A} \neq \emptyset$ because $\overline{A}$ is infinite.

Thus, if we choose (by the Fixed-Point Theorem) an e such that $\varphi_{h(e)} \simeq \varphi_e$, we must have $\mathcal{W}_{\varphi_e(x)} \cap \overline{A} \neq \emptyset$ for every x. Moreover, every $\mathcal{W}_{\varphi_e(x)}$ is finite, because $g(e,x,s) = 0$ only finitely often. $\square$

Exercise IX.4.18 *There are dense simple nonspeedable sets.* (Marques [1975], Soare [1977]) (Hint: if $\overline{A} = \{a_0 < a_1 < \cdots\}$, satisfy the obvious requirements

$$(\exists_\infty s)(\mathcal{W}_{e,s} \cap \overline{A}_s \neq \emptyset) \Rightarrow a_e \in \mathcal{W}_e,$$

so that $\mathcal{W}_e \cap \overline{A} \neq \emptyset$ and A is nonspeedable, and

$$a_e > \max\{\varphi_i(e) : \varphi_i(e)\downarrow \wedge\, i \leq e\},$$

so that A is dense simple.)

Obviously, only $\mathbf{0}'$ contains effectively speedable sets, since they are all Q-complete.

Proposition IX.4.19 (Gill and Morris [1974]) *Every strongly effectively simple set is effectively speedable.*

Proof. By IX.4.12 and III.6.21.a. $\square$

Exercises IX.4.20 a) *There are r-maximal, effectively speedable sets.* (Gill and Morris [1974]) (Hint: by III.4.11, a coinfinite r.e. set is hyperhypersimple if and only if it has no Q-complete superset. Thus any r-maximal, not hyperhypersimple set has a Q-complete superset, which must also be r-maximal.)

b) *There are dense simple, effectively speedable sets.* (Gill and Morris [1974]) (Hint: the semirecursive dense simple set constructed on p. 395 is T-complete because $\overline{A}$ dominates every partial recursive function, and hence is Q-complete by III.5.2.)

c) *No hyperhypersimple set is effectively speedable.* (Gill and Morris [1974], Soloviev [1974]) (Hint: by III.4.10.)

Complexity sequences ⋆

Meyer and Fisher [1972] have suggested, after looking at the proof of the Speed-Up Theorem VII.2.16, that the notion of a nonspeedable set may be extended in the following way: even if there is no single best complexity, there is an r.e. sequence of good complexities, cofinal in the complexities of the given set. More precisely, we say that $\{\Phi_{f(e)}\}_{e\in\omega}$ is an *r.e. complexity sequence for A* if:

- $A = \mathcal{W}_{f(e)}$ for every e
- for every index i of A there is e such that $(\forall_\infty x)[\Phi_{f(e)}(x) \leq \Phi_i(x)]$.

If the analogous notion is defined for functions, then the proof of the Speed-Up Theorem provides a function with an r.e. complexity sequence. But Bennison [1980] has proved that this notion is not measure-invariant: this is not surprising, since the step from one measure to another involves in general a cost function (see VII.1.15). We are thus led to the following notion.

Definition IX.4.21 (Blum) *An r.e. set A has a* **type-0 r.e. complexity sequence** *if there are recursive functions f and g such that:*

1. *$A = \mathcal{W}_{f(e)}$ for every e*
2. *for every index i of A there is e such that*

$$(\forall_\infty x)[\Phi_{f(e)}(x) \leq g(x, \Phi_i(x))].$$

This notion is now clearly measure-invariant, but the next result shows that it fails to provide an extension of the notion of nonspeedability.

Theorem IX.4.22 Characterization of the Sets with Type-0 Complexity Sequences (Bennison and Soare [1978]) *An r.e. set has a type-0 r.e. complexity sequence if and only if it is nonspeedable.*

Proof. If A is nonspeedable, then it has a single best complexity modulo some recursive function g, and thus a trivial type-0 r.e. complexity sequence.

If A has a type-0 r.e. complexity sequence, we can prove exactly as in the proof of IX.4.8 (in particular, with $\mathcal{W}_{h(i)} = A \cup \mathcal{W}_i$) that

$$\mathcal{W}_i \cap \overline{A} = \emptyset \Leftrightarrow (\exists n)(\exists e)(\forall x)[\Phi_{f(e)}(x) \leq g(x, \Phi_{h(i)}(x)) + n],$$

and thus $H_{\overline{A}} \in \Pi_2^0$. Then A is nonspeedable by IX.4.9. $\square$

Corollary IX.4.23 *An r.e. set A has a type-0 r.e. complexity sequence if and only if $\overline{A}$ is semilow.*

Since in this chapter we have systematically looked at properties that hold for almost every x, it is natural to look at the following extension of IX.4.21.

Definition IX.4.24 (Bennison and Soare [1978]) *An r.e. set A has a* **type-1 r.e. complexity sequence** *if there are recursive functions f and g such that:*

1. *$A =^* \mathcal{W}_{f(e)}$ for every e*
2. *for every index i such that $A =^* \mathcal{W}_i$ there is e such that*

$$(\forall_\infty x)[\Phi_{f(e)}(x) \leq g(x, \Phi_i(x))],$$

where $=^*$ *means 'equal with at most finitely many exceptions'.*

The proof of IX.4.22 suggests that we might obtain a characterization of the sets with type-1 r.e. complexity sequences by looking at the condition

$$\text{`}\mathcal{W}_i \cap \overline{A} \text{ finite' is } \Sigma^0_2,$$

since certainly any set with a type-1 r.e. complexity sequence has this property.

Exercises IX.4.25 Relativized Finiteness Problem. Let

$$F^A = \{x : \mathcal{W}^A_x \text{ finite}\}.$$

a) $F^A \equiv_1 A''$. (Hint: by relativization of X.9.6.)

b) *A is low if and only if $F^A \equiv_1 \mathcal{K}'$.* (Hint: $F^A \equiv_1 \mathcal{K}'$ if and only if F^A is Σ^0_2-complete. But F^A is $\Sigma^{0,A}_2$-complete, and $\Sigma^{0,A}_2 = \Sigma^0_2$ if and only if $\Sigma^{0,A'}_1 = \Sigma^{0,\mathcal{K}}_1$, i.e. if and only if A is low.)

c) *If $A \leq_T \mathcal{K}$, A is low$_2$ if and only if $F^A \equiv_T \mathcal{K}'$.* (Hint: from part a).)

From IX.4.25.a (see also X.9.6) we notice that the set

$$F = \{x : \mathcal{W}_x \text{ finite}\}$$

(called **Fin** in X.9.6) is recursively isomorphic to $\emptyset''$. The set F^A of the exercises above is its recursion-theoretical relativization, and hence an equivalent definition of the double jump operator. As already done for the Halting Problem, we now consider a set-theoretical relativization.

Definition IX.4.26 (Bennison and Soare [1978]) *The set*

$$F_A = \{x : \mathcal{W}_x \cap A \textit{ finite}\}$$

is called the **weak double jump of *A***.

Exercises IX.4.27 a) *If A is infinite, then $\mathcal{K}' \leq_1 F_A \leq_1 A''$.* (Hint: let $\mathcal{W}_{f(x)}$ be the downward closure of $\mathcal{W}_x$. Then

$$\mathcal{W}_x \text{ finite} \Leftrightarrow \mathcal{W}_{f(x)} \cap A \text{ finite},$$

and $\mathcal{K}' \leq_1 F_A$. Also, $F_A \in \Sigma^{0,A}_2$ and so $F_A \leq_1 A''$.)

b) *If A is r.e. and coinfinite then $\mathcal{K}' \leq_1 F_{\overline{A}} \leq_1 \mathcal{K}''$.* (Hint: if A is r.e. then $F_{\overline{A}} \in \Sigma^0_3$, and so $F_{\overline{A}} \leq_1 \mathcal{K}''$.)

Definition IX.4.28 (Bennison and Soare [1978]) *A set A is* **semilow$_{1.5}$** *if $F_A \equiv_1 \mathcal{K}'$ (or, equivalently for A infinite, $F_A \leq_1 \mathcal{K}'$).*

Exercises IX.4.29 A set A is **semilow$_2$** if $F_A \equiv_T \mathcal{K}'$.

a) *Semilow $\Rightarrow$ semilow$_{1.5}$ $\Rightarrow$ semilow$_2$, but the converse fails even for co-r.e. sets.* (Hint: if

$$\mathcal{W}_{f(x,z)} = \{y : y \in \mathcal{W}_x \ \wedge \ y \geq z\},$$

then

$$\mathcal{W}_x \cap A \text{ finite } \Leftrightarrow \ (\exists z)(\mathcal{W}_{f(x,z)} \cap A = \emptyset);$$

so if $H_A \leq_T, \mathcal{K}$ then $F_A \in \Sigma_2^0$ and $F_A \leq_1 \mathcal{K}'$. The first converse fails by IX.4.31, the second because there is a low$_2$ set which is not semilow$_{1.5}$, see part b) and IX.4.32.b.)

b) *Low$_2$ $\Rightarrow$ semilow$_2$, but the converse fails even for co-r.e. sets.* (Hint: $F_A \leq_1 A''$ by IX.4.27.a, and so if $A'' \equiv_T \mathcal{K}'$, then $F_A \equiv_T \mathcal{K}'$. The converse fails by IX.4.14 and part a).)

c) Notice that, in general, there cannot be a 0,1-valued recursive function g such that

$$\begin{array}{lcl} \mathcal{W}_x \cap \overline{A} \text{ finite} & \Rightarrow & \lim_{s\to\infty} g(x,s) = 1 \\ \mathcal{W}_x \cap \overline{A} \text{ infinite} & \Rightarrow & \lim_{s\to\infty} g(x,s) = 0, \end{array}$$

since otherwise $F_{\overline{A}} \leq_T \mathcal{K}$ (which, by IX.4.27.a, is impossible if A is coinfinite). The following provides a **weak version of the Limit Lemma**: *if A is r.e., then $\overline{A}$ is semilow$_{1.5}$ if and only if there is a 0,1-valued recursive function g such that*

$$\begin{array}{lcl} \mathcal{W}_x \cap \overline{A} = \emptyset & \Rightarrow & \lim_{s\to\infty} g(x,s) = 1 \\ \mathcal{W}_x \cap \overline{A} \textit{ infinite} & \Rightarrow & \lim_{s\to\infty} g(x,s) = 0. \end{array}$$

(Bennison and Soare [1978]) (Hint: if $F_{\overline{A}} \leq_1 \mathcal{K}'$ then $F_{\overline{A}}$ is r.e. in $\mathcal{K}$. Since A is r.e., $H_{\overline{A}}$ is r.e. in $\mathcal{K}$. Moreover, $F_{\overline{A}} \cup H_{\overline{A}} = \omega$. By the Reduction Property II.1.23 relativized to $\mathcal{K}$, there is a set $B \leq_T \mathcal{K}$ such that $B \subseteq F_{\overline{A}}$ and $\overline{B} \subseteq H_{\overline{A}}$. Let g be a function whose limit is c_B.

Conversely, given g there is B as above. Let

$$\mathcal{W}_{f(x,z)} = \{y : y \in \mathcal{W}_x \ \wedge \ y \geq z\}.$$

Then, from $\overline{H_{\overline{A}}} \subseteq B \subseteq F_{\overline{A}}$,

$$\mathcal{W}_x \cap \overline{A} \text{ finite } \Leftrightarrow \ (\exists z)(\mathcal{W}_{f(x,z)} \cap \overline{A} = \emptyset) \ \Leftrightarrow \ (\exists z)(f(x,z) \in B).$$

Since $B \leq_T \mathcal{K}$, $F_{\overline{A}}$ is r.e. in $\mathcal{K}$, hence $F_{\overline{A}} \leq_1 \mathcal{K}'$.)

Theorem IX.4.30 Recursion-Theoretic Characterization of the Sets with Type-1 Complexity Sequences (Bennison and Soare [1978]) *An r.e. set A has a type-1 r.e. complexity sequence if and only if $\overline{A}$ is semilow$_{1.5}$.*

Proof. If A has a type-1 r.e. complexity sequence, we can prove exactly as in the proof of IX.4.8 (in particular, with $\mathcal{W}_{h(i)} = A \cup \mathcal{W}_i$) that

$$\mathcal{W}_i \cap \overline{A} \text{ finite } \Leftrightarrow \ (\exists e)(\forall_\infty x)[\Phi_{f(e)}(x) \leq g(x, \Phi_{h(i)}(x))].$$

The right-hand side is Σ_2^0 (since $\forall_\infty x$ can be rewritten as $\exists y \forall x \geq y$), thus $F_{\overline{A}} \leq_1 \mathcal{K}'$ and $\overline{A}$ is semilow$_{1.5}$.

Suppose now $\overline{A}$ semilow$_{1.5}$. We proceed as in IX.4.8, and look for f and g recursive such that $A =^* \mathcal{W}_{f(e)}$ for every e, and

$$A =^* \mathcal{W}_i \quad \Rightarrow \quad (\exists e)(\forall_\infty x)[\Phi_{f(e)}(x) \leq g(x, \Phi_i(x))]. \tag{IX.1}$$

We first obtain recursive functions f and g_1 such that $A =^* \mathcal{W}_{f(e)}$ for every e, and

$$A = \mathcal{W}_e \quad \Rightarrow \quad (\forall_\infty x)[\Phi_{f(e)}(x) \leq g_1(x, \Phi_e(x))]. \tag{IX.2}$$

By the weak version of the Limit Lemma IX.4.29.c, there is a 0,1-valued recursive function h such that

$$\begin{array}{lll} \mathcal{W}_e \cap \overline{A} = \emptyset & \Rightarrow & \lim_{s \to \infty} h(e,s) = 1 \\ \mathcal{W}_e \cap \overline{A} \text{ infinite} & \Rightarrow & \lim_{s \to \infty} h(e,s) = 0. \end{array}$$

To obtain $\mathcal{W}_{f(e)} =^* A$, we make sure that we add only finitely many elements to A. When $A = \mathcal{W}_e$, we also want to have $\mathcal{W}_{f(e)} = A$, so it is enough to add to A only finitely many elements of $\mathcal{W}_e$:

$$\mathcal{W}_{f(e)} = A \cup \{x : x \in \mathcal{W}_e \ \wedge \ h(e,x) = 1\}.$$

Now $\mathcal{W}_{f(e)} =^* A$ because only finitely many elements of $\overline{A}$ go into $\mathcal{W}_{f(e)}$, either because $\mathcal{W}_e \cap \overline{A}$ is itself finite, or because (when $\mathcal{W}_e \cap \overline{A}$ is infinite) $\lim_{x \to \infty} h(e,x) = 0$.

To define g_1 we simply let

$$t_1(e,x,z) = \begin{cases} \Phi_{f(e)}(x) & \text{if } \Phi_e(x) \simeq z \ \wedge \ h(e,x) = 1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$g_1(x,z) = \max\{t_1(e,x,z) : e \leq x\}.$$

The function t_1 is well-defined because if $\Phi_e(x)\downarrow$ and $h(e,x) = 1$, then $\Phi_{f(e)}(x)\downarrow$ by definition of $\mathcal{W}_{f(e)}$. And if $A = \mathcal{W}_e$, then $\lim_{s \to \infty} h(e,x) = 1$, so for $x \geq e$ large enough we have

$$\Phi_{f(e)}(x) \leq g_1(x, \Phi_e(x)).$$

To show that A actually has a type-1 r.e. complexity sequence, i.e. to find g as in IX.1, it is enough to show that there is a recursive function g_2 such that, for every index i such that $A =^* \mathcal{W}_i$, there is an index e of A such that

$$(\forall_\infty x)[\Phi_e(x) \leq g_2(x, \Phi_i(x))]. \tag{IX.3}$$

Then, if g_1 is monotone in the second argument (which we may suppose, by possibly changing $g_1(x,z)$ into $\max_{y \leq z} g_1(x,y)$), by combining IX.2 and IX.3 we obtain that, for every index i such that $A =^* \mathcal{W}_i$, there is e such that

$$(\forall_\infty x)[\Phi_{f(e)}(x) \leq g_1(x, g_2(x, \Phi_i(x)))],$$

and thus f and $g = g_1 \circ g_2$ witness the fact that A has a type-1 r.e. complexity sequence.

Obtaining g_2 is immediate. First let $\mathcal{W}_{s(i,n)} = A \cup (\mathcal{W}_i - D_n)$, and then

$$t_2(i,n,x,z) = \begin{cases} \Phi_{s(i,n)}(x) & \text{if } \Phi_i(x) \simeq z \ \wedge \ x \notin D_n \\ 0 & \text{otherwise} \end{cases}$$

and

$$g_2(x,z) = \max\{t(i,n,x,z) : i \leq x \wedge n \leq x\}.$$

If $A =^* \mathcal{W}_i$, we then have $\mathcal{W}_i \cap \overline{A} = D_n$ for some n, hence $A = \mathcal{W}_{s(i,n)}$, and for $x \geq i, n$

$$\Phi_{s(i,n)}(x) \leq g_2(x, \Phi_i(x)),$$

so that $s(i,n)$ is the index e of A we were looking for. $\square$

Thus the r.e. sets with type-1 r.e. complexity sequences are characterized by the fact that their complements have the lowest possible weak double jump ($F_{\overline{A}} \equiv_1 \mathcal{K}'$). In this sense they are then analogues of the recursive sets, characterized in IX.4.5.d by the fact that their complements have the lowest possible weak jump ($H_{\overline{A}} \equiv_1 \mathcal{K}$).

At the opposite end of the spectrum, we do not know of any characterization of the r.e. sets whose complements have the highest possible weak double jump ($F_{\overline{A}} \equiv_1 \mathcal{K}''$), which would be analogues of the effectively speedable sets, characterized in IX.4.5.e (via IX.4.12) by the fact that their complements have the highest possible weak jump ($H_{\overline{A}} \equiv_1 \mathcal{K}'$). Bennison and Soare [1978] have shown that the former properly include the latter.

We conclude our treatment by discussing the relationships between the two notions of complexity sequences introduced so far.

Proposition IX.4.31 (Bennison and Soare [1978]) *If an r.e. set has a type-0 r.e. complexity sequence then it has a type-1 r.e. complexity sequence, but not conversely.*

Proof. The second part of the proof of IX.4.30 shows that the existence of a type-1 r.e. complexity sequence follows from the existence of recursive functions f and g_1 such that $A =^* \mathcal{W}_{f(e)}$ for every e, and

$$A = \mathcal{W}_i \ \ \Rightarrow \ \ (\exists e)(\forall_\infty x)[\Phi_{f(e)}(x) \leq g_1(x, \Phi_i(x))].$$

Thus any r.e. set with a type-0 r.e. complexity sequence also has a type-1 r.e. complexity sequence. Less directly, one can use IX.4.22, IX.4.8, IX.4.29.a and IX.4.30.

To show that the converse implication does not hold, we want to build an r.e. set A such that $\overline{A}$ is semilow$_{1.5}$ but not semilow.

- *$\overline{A}$ not semilow*
  To ensure $H_{\overline{A}} \not\leq_T \mathcal{K}$, notice that if $H_{\overline{A}} \leq_T \mathcal{K}$, then also
  $$\{x : \mathcal{W}_x \cap \overline{A} = \emptyset\} \leq_T \mathcal{K},$$
  and
  $$\{x : \mathcal{W}_x \cap \overline{A} = \emptyset\} \leq_1 \mathcal{K}' \equiv_1 F = \{x : \mathcal{W}_x \text{ finite}\}.$$
  Thus there is a total recursive function φ_e such that
  $$\mathcal{W}_x \cap \overline{A} = \emptyset \;\Leftrightarrow\; \mathcal{W}_{\varphi_e(x)} \text{ finite}.$$
  The requirements to make $\overline{A}$ not semilow are thus:
  $$P_e \;:\; (\exists x)(\mathcal{W}_x \subseteq A \;\Leftrightarrow\; \mathcal{W}_{\varphi_e(x)} \text{ infinite}).$$
  This is quite easily ensured. Since we can choose $\mathcal{W}_x$, we let it be
  $$\omega^{[e]} = \{\langle e, y\rangle : y \in \omega\}.$$
  Let x_e be an index of $\omega^{[e]}$: we ensure that
  $$\omega^{[e]} = \mathcal{W}_{x_e} \subseteq A \;\Leftrightarrow\; \mathcal{W}_{\varphi_e(x_e)} \text{ infinite}.$$
  The strategy for this is as follows: at stage $s+1$, if $\varphi_{e,s}(x_e)\downarrow$, then put into A all elements of $\omega^{[e]}$ smaller than the maximum element of $\mathcal{W}_{\varphi_e(x_e),s}$.

- *$\overline{A}$ semilow*$_{1.5}$
  For this we use the contrapositive of the weak version of the Limit Lemma IX.4.29.c, and define a 0,1-valued recursive function g such that
  $$\begin{array}{lcl} \lim_{s\to\infty} g(i,s) = 1 & \Rightarrow & \mathcal{W}_i \cap \overline{A} \text{ finite} \\ \lim_{s\to\infty} g(i,s) = 0 & \Rightarrow & \mathcal{W}_i \cap \overline{A} \neq \emptyset. \end{array}$$
  The strategy for this is as follows: we let $g(i,s) = 1$ until we find, for the first time, an element $z \in \mathcal{W}_{i,s} \cap \overline{A}_s$; then we permanently restrain z out of A (thus making $\mathcal{W}_i \cap \overline{A} \neq \emptyset$), and set $g(i,s)$ equal to 0 from then on (thus making $\lim_{s\to\infty} g(i,s) = 0$).

  If no such z is ever found, then $\lim_{s\to\infty} g(i,s) = 1$ and $\mathcal{W}_i \cap \overline{A}$ is empty (in particular, finite).

The strategy for the definition of g obviously conflicts with the strategy for the satisfaction of P_e if $z \in \omega^{[e]}$, but we can decide to satisfy P_e with just a part of $\omega^{[e]}$, say

$$\{\langle e, y\rangle : y > z\}.$$

We thus use a marker Γ_e, which moves along the e-th column above z, and tells us that if Γ_e^s is the position of Γ_e at stage s, then we are trying to satisfy P_e by using only the set

$$\mathcal{W}_{x_e^s} = \{\langle e, y\rangle : y > \Gamma_e^s\}.$$

Of course, we do not want to move Γ_e^s infinitely often, otherwise we are left with the empty set (which is automatically contained in A, and thus not a suitable witness for P_e). If we let the strategy for making $\overline{A}$ semilow$_{1.5}$ interfere with P_e only if $i \leq e$, then Γ_e^s moves (and x_e^s changes) only finitely often (at most once for every $i \leq e$), and the limits Γ_e of Γ_e^s and x_e of x_e^s exist.

Due to the condition $i \leq e$, it is not always possible to restrain elements out of $\overline{A}$ to ensure $\mathcal{W}_i \cap \overline{A} \neq \emptyset$, e.g. when $\mathcal{W}_i$ is contained in the first i columns (with indices $e < i$). In this case g is eventually equal to 1, and we want to ensure that $\mathcal{W}_i \cap \overline{A}$ is finite:

- We cannot simply avoid taking action since, for some $e < i$, it might be that $\mathcal{W}_i \cap \omega^{[e]}$ is infinite but $\mathcal{W}_{\varphi_e(x_e)}$ is finite: then the strategy for P_e puts only a finite subset of $\omega^{[e]}$ into A, and $\mathcal{W}_i \cap \overline{A}$ is infinite (thus disrupting the strategy for g).

- We also cannot put *all* elements of $\mathcal{W}_i \cap \omega^{[e]}$ above Γ_e into A, because it might be that $\mathcal{W}_i \cap \omega^{[e]} = \omega^{[e]}$, and then any subset of $\omega^{[e]}$ would be contained in A, even if $\mathcal{W}_{\varphi_e(x_e)}$ is finite (thus disrupting the strategy for P_e).

- It is however enough, at stage $s + 1$, to put into A_{s+1} each element of $\mathcal{W}_{i,s} \cap \omega^{[e]}$ greater than Γ_e^s, *except* for the first one which is not yet in A. This element may go into A later, because of the strategy for P_e, but if $\mathcal{W}_{\varphi_e(x_e)}$ is finite and $\mathcal{W}_i \cap \omega^{[e]}$ is infinite, then one element is permanently restrained out of A, and there is no conflict with the strategy for P_e.

  Since at most one element of $\mathcal{W}_i$ is permanently restrained out of A for each column $e < i$, $\mathcal{W}_i \cap \overline{A}$ is then finite.

The construction of A is as follows. Let g be a recursive enumeration of all pairs $\langle i, z\rangle$ such that $z \in \mathcal{W}_i$. At the end of (the first part of the construction at) stage s, we have Γ_e^s sitting on some element of $\omega^{[e]}$ (starting with $\Gamma_e^0 = \langle e, 0\rangle$), and x_e^s such that

$$\mathcal{W}_{x_e^s} = \{\langle e, y\rangle : \langle e, y\rangle > \Gamma_e^s\},$$

with the convention that x_e^s changes only when Γ_e^s moves. At stage $s+1$, we proceed as follows:

1. If $g(s) = \langle i, z\rangle$, go to part 2 unless $z \in \omega^{[e]} \cap \overline{A}_s$ and i has not yet been cancelled. In this case:
   - cancel i if $\langle i, z\rangle$ is below Γ_e^s (since the construction ensures $z \in \overline{A}$, and hence $\mathcal{W}_i \cap \overline{A} \neq \emptyset$)
   - if $i \leq e$ and $\langle i, z\rangle$ is not below Γ_e^s, move Γ_e to a position Γ_e^{s+1} above $\langle i, z\rangle$ and cancel i
   - if $i > e$ and $\langle i, z\rangle$ is not below Γ_e^s, put z in A unless $\langle i, z\rangle$ is the first element of $\omega^{[e]}$ above Γ_e^s which is not yet in A.
2. For $e \leq s$, if $\varphi_{e,s}(x_e^s)\downarrow$, then put all the elements of $\omega^{[e]}$ above Γ_e^s and smaller than the maximum element of $\mathcal{W}_{\varphi_e(x_e^s),s}$ into A.

Clearly Γ_e^s moves only finitely often, and so x_e^s has a limit x_e.

To prove that $\overline{A}$ is not semilow, we show

$$\mathcal{W}_{x_e} \subseteq A \;\Leftrightarrow\; \mathcal{W}_{\varphi_e(x_e)} \text{ infinite.}$$

The right-to-left direction is ensured by the second part of the construction. For the left-to-right direction, let $\mathcal{W}_{\varphi_e(x_e)}$ be finite, and let m_e be its maximum element. Then the first element $\langle e, z\rangle$ above Γ_e and m_e, not yet in A at a stage when Γ_e and m_e have attained their limits, and such that $z \in \mathcal{W}_e$, cannot enter A.

To prove that $\overline{A}$ is semilow$_{1.5}$, define

$$g(i,s) = \begin{cases} 1 & \text{if } i \text{ has not yet been cancelled at stage } s \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\lim_{s\to\infty} g(i,s) = 0 \;\Rightarrow\; \mathcal{W}_i \cap \overline{A} \neq \emptyset$$

by construction, and

$$\lim_{s\to\infty} g(i,s) = 1 \;\Rightarrow\; \mathcal{W}_i \cap \overline{A} \text{ finite}$$

because $\mathcal{W}_i \cap \overline{A}$ must be contained in the first i columns of ω (with indices $e < i$), otherwise we would cancel i, and at most one element of $\mathcal{W}_i$ above Γ_e (for $e < i$) can be permanently left out of A. $\square$

Exercises IX.4.32 Existence Theorems. (Bennison and Soare [1978])

a) *Every r.e. T-degree contains an r.e. set with type-1 r.e. complexity sequence.* (Hint: by IX.4.14, IX.4.22 and IX.4.31.)

b) *An r.e. T-degree contains an r.e. set which does not have a type-1 r.e. complexity sequence if and only if it is not low.* (Hint: in the notation of IX.4.16, we want to show $A'' \leq_1 F_{\overline{B}}$. Then if $F_{\overline{B}} \leq_1 \mathcal{K}'$, we have $A'' \leq_1 \mathcal{K}'$, i.e. $A' \leq_T \mathcal{K}$, and A is low.

Since $A'' \leq_1 F^A$, we want $F^A \leq_1 F_{\overline{B}}$, i.e. we want a recursive function g such that

$$\mathcal{W}_e^A \text{ finite } \Leftrightarrow \mathcal{W}_{g(e)} \cap \overline{B} \text{ finite.}$$

By the Normal Form Theorem for relativized r.e. sets (III.1.4), there is a recursive function f such that

$$x \in \mathcal{W}_e^A \Leftrightarrow (\exists u)(u \in \mathcal{W}_{f(x,e)} \wedge D_u \subseteq \overline{A}).$$

To define $g(x)$, wait for a stage s such that, for some u, $u \in \mathcal{W}_{f(x,e),s}$ and $D_u \subseteq \overline{A}_s$, and when this happens put u in $\mathcal{W}_{g(e)}$. If it later turns out that $D_u \cap A \neq \emptyset$, start again. Since

$$u \in \overline{B} \Leftrightarrow D_u \subseteq \overline{A},$$

it follows that

$$\mathcal{W}_e^A \text{ finite } \Leftrightarrow \mathcal{W}_{g(e)} \cap \overline{B} \text{ finite.)}$$

c) *No finitely strongly hypersimple set has a type-1 r.e. complexity sequence.* (Hint: modify the proof of IX.4.17 by using the weak version of the Limit Lemma IX.4.29.c, i.e. a recursive g such that

$$\begin{array}{lll} \mathcal{W}_{\varphi_e(x)} \cap \overline{A} = \emptyset & \Rightarrow & \lim_{s\to\infty} g(e,x,s) = 1 \\ \mathcal{W}_{\varphi_e(x)} \cap \overline{A} \text{ infinite} & \Rightarrow & \lim_{s\to\infty} g(e,x,s) = 0.) \end{array}$$

d) *There are dense simple sets with a type-1 r.e. complexity sequence.* (Hint: see IX.4.18.)

Maass [1983] has proved that for A coinfinite, $\mathcal{L}(A)$ *is effectively isomorphic to* $\boldsymbol{\mathcal{E}}$ *if and only if* $\overline{A}$ *is semilow*$_{1.5}$. In particular, from a lattice-theoretical point of view not only the nonspeedable sets, but also the r.e. sets with type-1 r.e. complexity sequences closely resemble the recursive sets.

IX.5 Inductive Inference of R.E. Sets ⋆

In Section VII.6 we developed a theory of inductive inference for total recursive functions. We now briefly indicate how this theory could be adapted to r.e. sets and partial recursive functions.

Identification by explanation of sets

In the framework of identification of sets, it makes no sense to consider an inference process that would guess the next value of an enumeration of the set, since only the range of an enumeration is important now, and not its order. We thus concentrate on analogues of the two notions of identification by explanation.

As for the recursive functions, the sets which are individually inferable are exactly the r.e. sets, and we thus turn our attention to classes of r.e. sets. The next definition is an analogue of VII.5.8 and VII.5.25.

Definition IX.5.1 (Gold [1967]) *Let $\mathcal{F}$ be a class of total functions. A class $\mathcal{C}$ of nonempty r.e. sets is* **identifiable by consistent $\mathcal{F}$-enumeration** *if there is a total recursive function g such that, for every sequence number $\langle a_0, \dots, a_n \rangle$:*

- $\{a_0, \dots, a_n\} \subseteq \mathcal{W}_{g(\langle a_0, \dots, a_n \rangle)}$,

and for every $f \in \mathcal{F}$ and (range f) $\in \mathcal{C}$:

- $\lim_{n \to \infty} g(\langle f(0), \dots, f(n) \rangle)$ *exists*
- $\mathcal{W}_{\lim_{n \to \infty} g(\langle f(0), \dots, f(n) \rangle)} =$ *range f.*

In other words, $g(\langle f(0), \dots, f(n) \rangle)$ provides a guess to an index of the range of f consistent with the available information, and the guess stabilizes (from a certain point on) on an index of the range of f.

$\mathcal{C}$ is **identifiable by $\mathcal{F}$-enumeration** *if the first condition on g is dropped.*

The next two results (which are analogues of VII.5.9 and VII.5.17, respectively) show that the notions above strongly depend on the class $\mathcal{F}$ of enumerating functions, and that there is a trade-off between the sizes of the classes $\mathcal{F}$ and $\mathcal{C}$. More precisely, if $\mathcal{F}$ is sufficiently small, then all r.e. sets are identifiable, and if $\mathcal{F}$ is sufficiently large, then nothing of interest is identifiable.

Proposition IX.5.2 (Gold [1967]) *The class of all (and hence any class of) nonempty r.e. sets is identifiable by consistent primitive recursive enumeration.*

Proof. By VII.5.9, any r.e. class $\mathcal{F} = \{\varphi_{h(e)}\}_{e \in \omega}$ of total recursive functions closed under finite variants is identifiable by consistent explanation, i.e. there is a total recursive function g such that, for every sequence number $\langle a_0, \dots, a_n \rangle$:

- $\varphi_{g(\langle a_0, \dots, a_n \rangle)}(x) \simeq a_x$ for all $x \leq n$,

and for every e:

- $\lim_{n \to \infty} g(\langle \varphi_{h(e)}(0), \dots, \varphi_{h(e)}(n) \rangle)$ exists
- $\varphi_{\lim_{n \to \infty} g(\langle \varphi_{h(e)}(0), \dots, \varphi_{h(e)}(n) \rangle)} = \varphi_{h(e)}$.

In particular, the first condition implies that

$$\{\varphi_{h(e)}(0), \dots, \varphi_{h(e)}(n)\} \subseteq \text{range } \varphi_{g(\langle \varphi_{h(e)}(0), \dots, \varphi_{h(e)}(n)\rangle)}.$$

By II.1.16, there is a recursive function t such that

$$\mathcal{W}_{t(e)} = \text{range } \varphi_e.$$

Then $t \circ g$ identifies by consistent $\mathcal{F}$-enumeration the class of all ranges of functions in $\mathcal{C}$.

It is now enough to notice the following. On the one hand, the class of primitive recursive functions is r.e. and closed under finite variants, and thus the class of the ranges of primitive recursive functions is identifiable by consistent primitive recursive enumeration. On the other hand, every nonempty r.e. set is the range of a primitive recursive function. Thus the class of all nonempty r.e. sets is identifiable by consistent primitive recursive enumeration. □

Proposition IX.5.3 (Gold [1967]) *No class of r.e. sets containing at least one infinite set and all its finite subsets is identifiable by recursive (let alone, arbitrary) enumeration.*

Proof. Let $\mathcal{C}$ be any such class, and g be a recursive function such that

$$f \text{ recursive} \wedge (\text{range} f) \in \mathcal{C} \;\Rightarrow\; \mathcal{W}_{\lim_{n\to\infty} g(\langle f(0), \dots, f(n)\rangle)} = \text{range } f.$$

By hypothesis, $\mathcal{C}$ contains an infinite r.e. set A, which is the range of a one-one recursive function h.

We define a total recursive function f whose range is still A, and such that $\lim_{n\to\infty} g(\langle f(0), \dots, f(n)\rangle)$ does not exist, contradicting the hypothesis on g. The idea is that f will repeat each value of h a number of times sufficient to force a change in g: since A is infinite, then g changes infinitely often on f, and thus it has no limit.

Start with $f(0) = h(0)$. Since $h(1) \neq h(0)$ because h is one-one, the two sets $\{h(0)\}$ and $\{h(0), h(1)\}$ are finite subsets of A (hence in $\mathcal{C}$), different, and the ranges of the two functions f_0 identically equal to $h(0)$, and f_1 equal to $h(0)$ for $x = 0$, and equal to $h(1)$ afterwards. By the hypothesis on g,

$$\lim_{n\to\infty} g(\langle f_0(0), \dots, f_0(n)\rangle) \neq \lim_{n\to\infty} g(\langle f_1(0), \dots, f_1(n)\rangle).$$

Thus there must exist n and i ($i = 0$ or $i = 1$) such that

$$g(\langle f(0)\rangle) \neq g(\langle f_i(0), \dots, f_i(n)\rangle),$$

and we can effectively find them. Let $f(x)$ be equal to $f_i(x)$ for every $x \leq n$, and to $h(1)$ for $x = n + 1$ (so that a new element has gone into the range of

f), and iterate the procedure (by enumerating, at each step, a new element of A into the range of f). $\Box$

The proofs of both results above exploit the possibility of repeating the same piece of information a great number of times. This is explicit in the proof of the last result, while in the first proof it is implicit in any primitive recursive enumeration of any highly complex r.e. set. Thus, it is the possibility of repetitions that makes the notion of identification by enumeration either trivial or drastically restricted.

From a different point of view, the two results just proved can be interpreted as saying that if we only allow for *positive information* in an inference process, then the process is sensitive to the order in which the data are made available, and it may become either trivially unrestricted or drastically restricted.

However, sets can be inferred not only through enumeration of their elements, but also through their characteristic functions, i.e. with *positive and negative information* available. In this case the results of the previous section apply, and the inference process does not collapse.

Exercises IX.5.4 rectify false impressions that one may get from IX.5.2 and IX.5.3: respectively, that (primitive) recursion is a necessary condition to guarantee identifiability of the class of all nonempty r.e. sets, and that no nontrivial class of r.e. sets is identifiable by recursive enumeration.

Exercises IX.5.4 (Wiehagen [1977]) a) *There is an uncountable class of functions $\mathcal{F}$, not containing any recursive function except for the constant ones, and such that the class of all nonempty r.e. sets is identifiable by consistent $\mathcal{F}$-enumeration.* (Hint: let $\mathcal{F}$ consist of all constant functions and, for any nonempty r.e. set A and some index e of A, of all the non recursive functions f with range $f = A$ and such that f is constant for all $x < e$, and changes value first at e. Then one identifies all nonempty r.e. sets by $\mathcal{F}$-enumeration, via g defined as follows: $g(\langle a_0, \dots, a_n\rangle)$ is an index of the constant function with value a_0 if $(\forall i \leq n)(a_i = a_0)$, and e if $(\forall i < e \leq n)(a_i = a_0)$ but $a_e \neq a_0$.)

b) *There is a class of r.e. sets identifiable by recursive (even arbitrary) enumeration, and containing a finite variant of every r.e. sets.* (Hint: consider the class

$$\mathcal{C} = \{\mathcal{W}_e : e \text{ is the least element of } \mathcal{W}_e\}.$$

If $g(\langle a_0, \dots, a_n\rangle) = \min\{a_0, \dots, a_n\}$, then g identifies $\mathcal{C}$.)

Exercises IX.5.5 Other notions of identification for sets. (Osherson and Weinstein [1982]) Let $\boldsymbol{EX_s}$ be the class of all classes of r.e. sets identifiable by arbitrary enumeration, and define $\boldsymbol{EX_s^*}$, $\boldsymbol{BC_s}$ and $\boldsymbol{BC_s^*}$ by adapting VII.5.38, VII.5.44 and VII.5.53 in a similar way.

a) *$EX_s \not\subseteq BC_s$, and thus the analogue of VII.5.47 fails.*[2] (Hint: consider the class $\mathcal{C}$ of all coinfinite r.e. sets. Obviously, $\mathcal{C} \in EX_s^*$ via the function that always outputs an index of ω. To show $\mathcal{C} \notin BC_s$, it is enough to show that if a recursive function g behaviorally correctly identifies a class $\mathcal{C}$ of r.e. sets, i.e. for all $A \in \mathcal{C}$ and f such that range $f = A$,

$$(\exists n_0)(\forall n \geq n_0)[\mathcal{W}_{g(\langle f(0),\ldots,f(n)\rangle)} = A],$$

then: for any $A \in \mathcal{C}$ and f such that range $f = A$,

$$(\exists n_0)(\forall n \geq n_0)[\mathcal{W}_{g(\langle f_1(0),\ldots,f_1(n)\rangle)} = A]$$

for any function f_1 that enumerates a subset of A containing $\{f(0),\ldots,f(n_0)\}$. Intuitively, this holds because if g converges to indices of A, then it does so on the basis of a finite amount of positive information about it, which cannot distinguish a subset of A from the whole set. Technically, if the property failed, then it would be possible to build an (arbitrary) enumeration of A on which, infinitely often, g takes values which are not indices of A; see also the first part of the proof of IX.5.9.)

b) *$BC_s \not\subseteq EX_s^*$.* (Hint: consider the class

$$\mathcal{C} = \{(\mathcal{K} \cup D) \cdot N : D \text{ finite}\},$$

i.e. the class of cylinders of finite variants of $\mathcal{K}$. Obviously, $\mathcal{C} \in BC_s$ via the function g that on $\langle a_0,\ldots,a_n\rangle$ takes as value an index of $(\mathcal{K} \cup \{b_0,\ldots,b_n\}) \cdot N$, where b_i is the first component of a_i. To show $\mathcal{C} \notin EX_s^*$, suppose g identifies $\mathcal{C}$ by explanation with finite errors, and f enumerates $\mathcal{K} \cdot N$: as in part a),

$$\begin{aligned}(\exists n_0)(\mathcal{W}_{g(\langle f(0),\ldots,f(n_0)\rangle)} =^* \mathcal{K} \cdot N \text{ and}\\ (\forall n \geq n_0)[g(\langle f(0),\ldots,f(n_0)\rangle) = g(\langle f_1(0),\ldots,f_1(n)\rangle)])\end{aligned}$$

for any function f_1 that enumerates a subset of $\mathcal{K} \cdot N$ containing $\{f(0),\ldots,f(n_0)\}$.

Given x, consider an enumeration f_x of $(\mathcal{K} \cup \{x\}) \cdot N$ such that $f_x(n) = f(n)$ for all $n \leq n_0$. Then

$$x \in \overline{\mathcal{K}} \Leftrightarrow (\exists n \geq n_0)[g(\langle f(0),\ldots,f(n_0)\rangle) \neq g(\langle f_x(0),\ldots,f_x(n)\rangle)],$$

and thus $\overline{\mathcal{K}}$ would be r.e.)

c) *$EX_s \subset EX_s^* \subset BC_s^*$.* (Hint: the inclusions are obvious, and they are proper by parts a) and b).)

d) *$EX_s \subset BC_s \subset BC_s^*$.* (Hint: as for part c).)

For more on identification of classes of r.e. sets, see Feldman [1972], Wiehagen [1977], Angluin [1980], Jantke and Beik [1981], Osherson and Weinstein [1982], Osherson, Stob and Weinstein [1982], [1982a], [1986], Case and Lynes [1982], Fulk [1990], Jantke [1991a].

[2]The analogue of the proof of VII.5.47 would eventually put in the finitely many elements left out by a wrong guess, but would not take out the finitely many elements wrongly put in by it.

Identification by explanation of partial functions

By the Graph Theorem II.1.11, the partial recursive functions are exactly the partial recursive functions with r.e. graph. One could thus think of identifying classes of partial recursive functions by identifying classes of r.e. graphs. However, IX.5.2 and IX.5.3 show that some care is needed, if one wants to avoid a collapse of the notion of identification by explanation.

While IX.5.2 is a positive results that seems unavoidable, there is hope of circumventing the negative result IX.5.3 by an appropriate weakening of the notion of identification. This is done in the following variation of IX.5.1, where one requires the identification not of the partial functions whose graphs are given as inputs (as done in Section VII.5), but only of extensions of them.

Since a partial recursive function can be undefined on some arguments, a sequence of numbers cannot be interpreted uniquely as a sequence of values (as we did in Section VII.5), without specifying the arguments they refer to. In the following we thus approximate partial functions by *two* finite sequences of numbers $\langle a_0, \ldots, a_n\rangle$ and $\langle b_0, \ldots, b_n\rangle$, where b_i is interpreted as the value of a function for the argument a_i. Since the domain of a partial recursive function is not necessarily recursive, by II.1.17 it cannot in general be recursively enumerated in increasing order, and thus the a_i's cannot in general be ordered by magnitude. The only condition the two sequences have to satisfy, in order to be considered as approximating a partial function, is that they are consistent in assigning the same value to the same argument, i.e.

$$(\forall i, j \leq n)(a_i = a_j \ \Rightarrow\ b_i = b_j).$$

Definition IX.5.6 (Blum and Blum [1975]) *Let $\mathcal{F}$ be a class of total functions. A class $\mathcal{C}$ of partial recursive functions is* **identifiable by $\mathcal{F}$-enumeration** *if there is a total recursive function g such that, for every $\varphi \in \mathcal{C}$ and $f \in \mathcal{F}$ such that range f = domain φ:*

- $\lim_{n\to\infty} g(\langle f(0), \ldots, f(n)\rangle, \langle \varphi(f(0)), \ldots, \varphi(f(n))\rangle)$ *exists*
- $\varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n)\rangle,\langle \varphi(f(0)),\ldots,\varphi(f(n))\rangle)} \supseteq \varphi$.

In other words, $g(\langle f(0), \ldots, f(n)\rangle, \langle \varphi(f(0)), \ldots, \varphi(f(n))\rangle)$ provides a guess to an index of a partial recursive extension of φ, and the guess stabilizes (from a certain point on) on an index of such an extension.

Before we go on, we notice that a similar patch-up of the notion of identification by explanation for classes of r.e. sets would not lead anywhere: ω contains every r.e. set, and it would thus be enough to always output an index of ω to identify the class of all r.e. sets in this weaker sense.[3]

[3]For the same reason, IX.5.3 cannot be circumvented by identifying r.e. classes of r.e. sets

As in IX.5.2 and IX.5.3, we now concentrate on the two cases of primitive recursive and recursive enumerations. As anticipated, the analogue of IX.5.2 continues to hold.

Proposition IX.5.7 (Gold [1967], Blum and Blum [1975]) *The class of all (and hence any class of) partial recursive functions is identifiable by primitive recursive enumeration.*

Proof. Similar to IX.5.2. □

The good news is that the analogue of IX.5.3 instead fails, in a strong sense.

Proposition IX.5.8 (Blum and Blum [1975]) *Any class of total recursive functions in EX is identifiable (as a class of partial functions) by arbitrary enumeration, and hence also by recursive enumeration.*

Proof. Definition VII.5.25 is a special case of IX.5.6: $\mathcal{F}$ consists only of the identity function or, equivalently, graphs of functions are canonically enumerated in the order by magnitude of the arguments. The idea is simply to extract such a canonical enumeration from any enumeration of the graph, using the fact that the latter must eventually enumerate all arguments, if the function is total.

If $\mathcal{C}$ is identified by explanation as a class of total function via g, it is identified as a class of partial function by arbitrary enumeration via g_1 defined as follows. Given $\langle a_0, \ldots, a_n \rangle$ and $\langle b_0, \ldots, b_n \rangle$, let

$$g_1(\langle a_0, \ldots, a_n \rangle, \langle b_0, \ldots, b_n \rangle) = g(\langle b_{i_0}, \ldots, b_{i_m} \rangle),$$

where $a_{i_0}, \ldots, a_{i_m}$ is the longest subsequence of $a_0, \ldots, a_n$ defined as follows: a_{i_0} is the first a_i which is equal to 0, a_{i_1} the first a_i which is equal to 1, and so on. If there is no such subsequence, let

$$g_1(\langle a_0, \ldots, a_n \rangle, \langle b_0, \ldots, b_n \rangle) = 0.$$

If $f \in \mathcal{C}$, then g reaches a correct limit on a sufficiently long segment of the canonical enumeration of the graph, and hence so does g_1 on any enumeration, as soon as that segment has been generated by it. □

In Definition IX.5.6 the guesses depend not only extensionally on the graphs of the input functions, but also intensionally on their enumerations, and hence

as in the proof of VII.5.9. As soon as an index of an infinite r.e. set in a given class is reached, then the guess will always be consistent with the information about any of its r.e. subsets in the same class.

so do their limits. Since these only have to converge on indices of partial recursive *extensions* of the input functions, the possibility arises that different extensions are obtained for different enumerations of the graph of the same input function. The next result shows that one can restrict attention to procedures for which this does not happen, in a strong sense.

Proposition IX.5.9 (Blum and Blum [1975]) *A class of partial recursive functions identifiable by recursive enumeration is also identifiable by arbitrary enumeration, independently of the enumeration.*

Proof. Let $\mathcal{C}$ be a class of partial recursive functions, identified by recursive enumeration via g. The proof proceeds in two steps: first one shows that, for any $\varphi \in \mathcal{C}$ and any recursive enumeration f of its domain, there is a segment of the enumeration on which g reaches a limit relatively not only to that enumeration, but to any enumeration extending that segment; then one recursively chooses a unique such segment, for example the first one in some recursive order of them.

For the first part, let $\varphi \in \mathcal{C}$ and a recursive enumeration f of its domain be given, and suppose for the sake of contradiction that for every segment of f there is a segment of some enumeration of the domain of φ extending it and on which g changes value. We define by induction a recursive enumeration of the graph of φ on which g changes infinitely often and hence does not reach a limit, thus contradicting the hypothesis that g identifies $\mathcal{C}$ by recursive enumeration. To do this, it is enough at each stage: first, to insert a new element of the graph of φ (to eventually really obtain an enumeration of it); second, to search for an enumeration of a finite portion of the graph of φ extending the current one, and making g change value. The latter search can be made effective by comparing all enumerations of graphs of finite functions with a recursive enumeration of the graph of φ (obtained by computing φ on the elements of the range of f): this produces all enumerations of graphs of finite subfunctions of φ, on which g can be computed.

For the second part, one proceeds in a way similar to that in IX.5.8, by defining g_1 as follows. Given $\langle a_0, \ldots, a_n \rangle$ and $\langle b_0, \ldots, b_n \rangle$, let

$$g_1(\langle a_0, \ldots, a_n \rangle, \langle b_0, \ldots, b_n \rangle) = g(\langle a_{i_0}, \ldots, a_{i_m} \rangle, \langle b_{i_0}, \ldots, b_{i_m} \rangle),$$

where $\langle a_{i_0}, \ldots, a_{i_m} \rangle$ is the smallest number of a subsequence of $\langle a_0, \ldots, a_n \rangle$ such that $g(\langle a_{i_0}, \ldots, a_{i_m} \rangle, \langle b_{i_0}, \ldots, b_{i_m} \rangle)$ does not change on any other subsequence extending it. $\square$

For more on identification of classes of partial recursive functions, see Blum and Blum [1975], Minicozzi [1976], Case and Smith [1983].

æ

Chapter X

Recursively Enumerable Degrees

In Chapter IX we looked at recursively enumerable sets from a lattice-theoretical point of view. We now look at them from the point of view of relative computability and study their degree structure following the paradigm set up in Chapter V and Chapter VI.

The first four sections of the chapter deal with methodology, and the first three correspond to Sections 2–4 of Chapter V. We start in Section 1 with a constructive analogue of the finite extension method, called the **finite injury priority method**. In Section 2 we show that the finite injury priority method is a constructive version of the Baire category method. In Section 3 we introduce the **infinite injury priority method**, that provides an analogue of the coinfinite extension method. In Section 4 we deal with priority arguments in general, by introducing the **tree method** and discussing various kinds of solutions to Post's Problem.

The remaining sections of the chapter provide a systematic treatment of various structures of r.e. degrees. We deal with ***m*-degrees** in Section 5, with **Turing degrees** and ***wtt*-degrees** in Section 6, and with **1-degrees**, ***btt*-degrees** and ***tt*-degrees** in Section 7. Since we cannot develop all these theories completely, we concentrate on m-degrees and T-degrees, but we do develop the remaining theories up to a point where we can show that they are not pairwise elementarily equivalent. In Section 8 we deal with **structure inside r.e. degrees**, along the lines of Section VI.6. Finally, in Section 9 we treat the m-degrees and the T-degrees of **index sets**.

We do not discuss in this chapter the behavior of the jump operator on r.e. sets, since we deal with it in more generality in Chapter XI.

Before we start, we define the structures we will look at in the chapter.

Definition X.0.10 *For any reducibility $\leq_r$ between $\leq_1$ and $\leq_{wtt}$, $\boldsymbol{\mathcal{R}_r}$ is the substructure of $\boldsymbol{\mathcal{D}_r}$ consisting of the r.e. r-degrees.*

$\boldsymbol{\mathcal{R}}$ is the substructure of $\boldsymbol{\mathcal{D}}$ consisting of the r.e. T-degrees.

X.1 The Finite Injury Priority Method

We have devoted a good deal of Chapter III to a solution to Post's Problem. The one we gave was not only a solution, it was also in the spirit of Post's paper [1944]. But these were later developments, which culminated in Marchenkov [1976]. A solution by brute force was found earlier by Muchnik [1956] and Friedberg [1957a], who introduced the finite injury priority method and constructed directly an r.e. set that is neither recursive nor T-complete. We begin this chapter by giving two different proofs of this type of solution to Post's Problem.

We want to build an r.e. set A which is nonrecursive and T-incomplete. The first part is easy, e.g. we can try to make A simple (as in III.2.11). To make A not T-complete we have two methods:

- A is T-complete if $B \leq_T A$ for every r.e. set B. So it is enough to build, together with A, an r.e. set B such that $B \not\leq_T A$.
- Equivalently (III.1.3), A is T-complete if $\mathcal{K} \leq_T A$. So it is also enough to force $\mathcal{K} \not\leq_T A$.

The experience of Chapter V tells us that the first method is easier (since we have complete control over both A and B), while the second method would require some insight (for example, in the non-r.e. case, the splitting method of V.2.12). We thus begin with the first approach.

The finite injury method is not difficult in theory, but it is cumbersome to write out complete proofs of results proved by it. We will now spell out in detail the reasons why we proceed in a certain way (by relying on some of the work done in Section V.2), while in future proofs we will just outline the required strategies, in many cases leaving to the reader the task of completing the formal proof.

Motivation

We want to build two r.e. sets A and B such that A is coinfinite and the following requirements are satisfied:

$$\begin{array}{rcl} P_e & : & \mathcal{W}_e \text{ infinite } \Rightarrow \mathcal{W}_e \cap A \neq \emptyset \\ N_e & : & B \not\simeq \{e\}^A. \end{array}$$

The first task we have to undertake is to ensure that the construction really gives r.e. sets. To achieve this, the construction must be recursive (since it has to recursively enumerate sets) and it will define at stage s a recursive set A_s, uniformly in s. Then $A = \bigcup_{s\in\omega} A_s$ is r.e., since

$$x \in A \Leftrightarrow (\exists s)(x \in A_s).$$

The main difference with the constructions of Section V.2 is that at a certain stage we can decide to put a certain element into A, but not into $\overline{A}$. If we want to have some element in $\overline{A}$, the best we can do is to try to be clever enough to avoid putting it into A forever. Similarly for B.

The letter P in P_e indicates that this is a **positive requirement**, since to satisfy it we want to put some element (of $\mathcal{W}_e$) into A. A crucial property of P_e is that it just requires a finite action (only one element in this case) for satisfaction. An accidental property is that once P_e is satisfied, it remains so forever (because, as $A = \bigcup_{s\in\omega} A_s$, we are not allowed to take elements out of A that are already in). In figurative language, we say that P_e cannot be **injured**.

The letter N in N_e indicates that this is a **negative requirement** for A (the basic set we are constructing), since to satisfy it we want to keep some elements out of A, as we now explain. As in Section V.2, N_e will be satisfied if, for some element z_e, $B(z_e) \not\simeq \{e\}^A(z_e)$. Suppose at a certain stage s we pick $z_e \notin A_s$ as our witness (some authors say that we appoint z_e as a **follower** to N_e). Since our construction has to be recursive, we cannot ask if there is a string σ such that $\{e\}^\sigma(z_e)\downarrow$ (this question is only recursive in $\mathcal{K}$). What we can do is to ask, at any given stage $t \geq s$, if $\{e\}_t^{A_t}(z_e)\downarrow$. So we wait until we find such a stage t, if ever. If we never find it, then $\{e\}^A(z_e)$ does not converge and thus $B \not\simeq \{e\}^A$. If we do find it, we would like to fix the computation, i.e. ensure that $\{e\}^A(z_e) \simeq \{e\}_t^{A_t}(z_e)$.

Proposition X.1.1 Preserving Computations Lemma. *Let $\{A_s\}_{s\in\omega}$ be a nondecreasing sequence of (not necessarily recursive) sets, and let $A = \bigcup_{s\in\omega} A_s$. If*

1. *$\{e\}_s^{A_s}(x) \simeq y$ and the computation queries the oracle only for elements smaller than u*

2. *$(\forall z < u)(z \notin A_s \Rightarrow z \notin A)$*

then

3. *$(\forall t \geq s)(\{e\}_t^{A_t}(x) \simeq y)$ and the computation queries the oracle only for elements smaller than u*

4. *$\{e\}^A(x) \simeq y$.*

Proof. If $z < u$, then either $z \in A_s$, and then $z \in A$ because $A_s \subseteq A$, or $z \notin A_s$, and then $z \notin A$ by hypothesis 2. Then the computation from the oracle A is the same as the computation from the oracle A_s.

Similarly, for any $t \geq s$, either $z \in A_s$, and then $z \in A_t$ because $A_s \subseteq A_t$, or $z \notin A_s$, and then $z \notin A$ by hypothesis 2, and hence $z \notin A_t$. $\square$

To fix the computation $\{e\}_t^{A_t}(z_e)$ it is thus enough to keep out of A forever the elements used in the computation and not in A_t (we call them elements used negatively). We thus put them aside into an A-box, and ask that they never be put into A. Moreover, if $\{e\}_t^{A_t}(z_e) \not\simeq 0$, we want z_e to stay out of B (so we put it into a B-box), otherwise we put z_e into B. If our wishes are all satisfied, at the end

$$z_e \in B \;\Leftrightarrow\; \{e\}^A(z_e) \simeq \{e\}_t^{A_t}(z_e) \simeq 0.$$

Note that the action for N_e only has a negative effect on A, but can have a positive or a negative effect on B, depending on the outcome of the computation.

To avoid putting unwanted elements of the B-box into B is very easy. It is enough to choose different witnesses for different requirements, so that there is no interference.

To avoid putting elements of the A-box into A is tricky, since we also want to satisfy the positive requirements. Suppose, for example, that at a certain stage $s+1$ we have A_s, B_s and the current A-box, we see $\mathcal{W}_{e,s} \cap A_s = \emptyset$ (so that P_e has not yet been satisfied), and there is $x \in \mathcal{W}_{e,s}$ such that $x \geq 2e$. If x is not in the A-box we can put it into A, thus satisfying P_e. What if x is in the A-box? On the one hand we would like to put it into A to satisfy P_e; on the other hand we would like to leave it out of A to preserve some negative requirement N_i, since this is the reason why an element gets into the A-box.

A moment of reflection shows that the prudent strategy of always giving precedence to the negative requirements cannot work, since it might be that, for some e, $\mathcal{W}_e$ is infinite but is generated so slowly that, whenever we discover an element that would be good to satisfy P_e, that element is already in the A-box. Thus P_e would never obtain satisfaction.

It is then necessary, from time to time, to take an element out of the A-box and put it into A. Suppose this element fell into the box because we wanted to preserve $\{i\}_t^{A_t}(z_i)$ for some i and t. By putting it into A we change the oracle of this computation, so that we are no longer sure that $\{i\}_t^{A_t}(z_i) \simeq \{i\}^A(z_i)$ and z_i is no longer a trustable witness of $B \not\simeq \{i\}^A$. We say that requirement N_i has been **injured**, and we will have to pick up a new witness and start all over again.

Another moment of reflection shows that the bold strategy of always giving precedence to the positive requirements does not work either, since it might be that, for some i, every time we want to preserve a computation for some witness, there comes a positive requirement injuring N_i. The problem is that infinitely many injuries may kill N_i (see Section 3 for conditions under which N_i survives).

What do we do then? Well, we do not have to be too radical. We decide to order the requirements in a **priority list** of length ω, e.g.

$$N_0 > P_0 > N_1 > P_1 > \cdots$$

(although any other would do), and when conflicts arise we give way to the requirement of highest priority. Since for every requirement there are only finitely many with higher priority, and since only finite action is needed to satisfy the positive ones, eventually we can satisfy all the requirements.

Note that also the idea of trying to satisfy the positive requirements before the negative ones (or conversely) amounted to putting the requirements in a priority list. However, the list was a complicated one (of length $\omega + \omega$) and did not work. In Generalized Recursion Theory there is often the need to simplify bad priority lists in a similar way. Also, priority lists were used repeatedly in Chapter III (e.g. in the construction of simple, hypersimple and maximal sets, see III.2.11, III.3.12, and III.4.18), and throughout Chapter VII (see VII.2.3, VII.2.16, VII.3.6, and VII.3.7).

Embeddability results

We are now going to solve Post's Problem again, by implementing the ideas discussed above.

Theorem X.1.2 (Muchnik [1956], Friedberg [1957a]) *There exists an r.e. set which is neither recursive nor T-complete.*

Proof. We begin by letting $A_0 = B_0 = \emptyset$. We choose our witnesses for N_e on the e-th column of ω (thought of as a set of pairs), so that they do not interfere with the witnesses for other negative requirements. Let

$$z_e^0 \;=\; \text{the first element of the } e\text{-th column} \;=\; \langle e, 0\rangle.$$

At stage $s+1$, we have A_s, B_s, z_e^s and the following functions:

$$u(e,x,s) = \begin{cases} \mu y.\,(\{e\}_s^{A_s[y]}(x)\downarrow) & \text{if it exists} \\ 0 & \text{otherwise,} \end{cases}$$

where $A_s[y] = A_s \cap \{0, \ldots, y\}$ (u is called the **use function**, and gives a bound on the elements used in the computation $\{e\}_s^{A_s}(x)$, when it converges), and

$$r(e,s) \;=\; u(e, z_e^s, s)$$

(r is called the **restraint function**, and gives a bound on the elements we want to keep out of A to preserve the computation $\{e\}_s^{A_s}(z_e^s)$ when it converges, to avoid injuries to N_e).

The construction at step $s+1$ is as follows:

- *B-part*
  For each $e \leq s$, if $\{e\}_s^{A_s}(z_e^s) \simeq 0$ and $z_e^s \notin B_s$ (i.e. N_e is not yet satisfied), put z_e^s into B.

- *A-part*
  For each $e \leq s$, if $\mathcal{W}_{e,s} \cap A_s = \emptyset$ (i.e. P_e is not yet satisfied) and, for some x,
  $$x \in \mathcal{W}_{e,s} \ \wedge \ x \geq 2e \ \wedge \ (\forall i \leq e)(x > r(i,s))$$
  (i.e. there is a suitable element in $\mathcal{W}_e$ which is not restrained by conditions of higher priority), then take the least such x and put it into A.

We say that N_e has been **injured at step $s+1$** if for some x

$$x \in A_{s+1} - A_s \ \wedge \ x \leq r(e,s)$$

(i.e. if we put in A an element that we wanted to keep outside).

The new witnesses are now:

$$z_e^{s+1} = \begin{cases} z_e^s & \text{if } N_e \text{ has not been injured} \\ & \text{at step } s+1, \text{ or } z_e^s \notin B_{s+1} \\ \mu x.\,(\exists z \leq x)(\langle e,z\rangle = x \ \wedge \ x > z_e^s) & \text{otherwise.} \end{cases}$$

Note that z_e^s does not change when it is not in B_{s+1}, even if N_e has been injured, since we can still used it as a witness. This is not necessary, but it allows us to prove X.1.5 below.

It is clear that $A = \bigcup_{s\in\omega} A_s$ and $B = \bigcup_{s\in\omega} B_s$ are r.e., and $\overline{A}$ is infinite (by the condition $x \geq 2e$).

- *N_e is satisfied, for each e*
  Note that N_e can be injured only finitely many times: to injure it at stage $s+1$, there must be $x \in A_{s+1} - A_s$ such that $x \leq r(e,s)$, hence a positive condition with index $< e$ has been satisfied at stage $s+1$. But there are only finitely many such conditions, and each can be satisfied at most once. Let s_0 be such that
  $$(\forall s \geq s_0)(N_e \text{ is not injured at stage } s+1),$$
  and $z_e = \lim_{s\to\infty} z_e^s = z_e^{s_0}$.

If $\{e\}^A(z_e)$ does not converge, then N_e is satisfied because $\{e\}^A$ is not total.

If $\{e\}^A(z_e)$ converges, let s_1 be the first stage greater than s_0 at which $\{e\}_{s_1}^{A_{s_1}}(z_e)$ converges. Since N_e is not injured after s_0, this is the final computation. By construction

$$z_e \in B \;\Leftrightarrow\; \{e\}^A(z_e) \simeq 0,$$

so $B \not\simeq \{e\}^A$.

- $\lim_{s\to\infty} r(e,s) < \infty$, *for each* e
  With the notation used above,

  $$\lim_{s\to\infty} r(e,s) = \begin{cases} 0 & \text{if } \{e\}^A(z_e) \text{ does not converge} \\ u(e, z_e, s_1) & \text{otherwise} \end{cases}$$

  because, after s_0, if $\{e\}_s^{A_s}(z_e^s)$ converges, then it is preserved forever.

- P_e *is satisfied, for each* e
  Let

  $$\begin{aligned} r(e) &= \lim_{s\to\infty} r(e,s) \\ R(e) &= \max_{i\le e} r(i), \end{aligned}$$

  and s_2 be a stage such that

  $$(\forall s \ge s_2)(\forall i \le e)[r(i,s) = r(i,s_2)].$$

  If $\mathcal{W}_e$ is infinite, there is an x such that

  $$x \in \mathcal{W}_e \;\wedge\; x \ge 2e \;\wedge\; x > R(e).$$

  So either $\mathcal{W}_{e,s_2} \cap A_{s_2} \ne \emptyset$ or, as soon as one of those x's appears in $\mathcal{W}_e$ after stage s_2, it goes into A. $\square$

Following Rogers [1967], we may visualize the behavior of z_e^s by using **movable markers** Γ_e associated to the integers z_e^s at step s. When we change witness, we move the corresponding marker. Since $\lim_{s\to\infty} z_e^s$ exists, each marker eventually settles down on the final witness.

The **injury set**

$$I_e = \{x : (\exists s)[x \in A_{s+1} - A_s \;\wedge\; x \le r(e,s)]\},$$

which is always r.e. by definition, is in this case finite. In Section 3 we will learn how to handle infinite injury sets.

Exercise X.1.3 *The auxiliary set B is not simple.* (Rogers [1967]) (Hint: consider a recursive sequence e_n of indices of the completely undefined function. Then $z_{e_n} = z^0_{e_n}$ never goes into B.)

It may be instructive to take a closer look at the function z_e which gives the witness for N_e. This is the crucial point to focus on, since it embodies (through its approximations) the heart of the priority method used in the proof.

Proposition X.1.4 (Friedberg [1957a], Soare [1972]) *z_e is not a recursive function, but it is recursive in $A \oplus B$.*

Proof. By construction, we have

$$z_e \in B \;\Leftrightarrow\; \{e\}^A(z_e) \simeq 0.$$

Let g be a recursive function such that

$$\{g(e)\}^A(x) \simeq 0 \;\Leftrightarrow\; e \in \overline{A},$$

so that $\{g(e)\}^A$ is total if $e \in \overline{A}$, and completely undefined otherwise. Then

$$e \in \overline{A} \;\Leftrightarrow\; \{g(e)\}^A(z_{g(e)}) \simeq 0 \;\Leftrightarrow\; z_{g(e)} \in B.$$

Since B is r.e., if z_n were a recursive function of n, then $\overline{A}$ would be r.e., contradicting X.1.2.

By construction, z_e is either the greatest element of the e-th column which is in B, or the smallest one which is not in B (since $z^{s+1}_e \neq z^s_e$ only if $z^s_e \in B_{s+1}$). These two elements, call them $z_{e,1}$ and $z_{e,0}$, are easily found recursively in B. It remains to decide which one is the true z_e. Since $z_{e,1} \in B$, it was put in B because at a certain stage s it was $\{e\}^{A_s}_s(z_{e,1}) \simeq 0$. We can recursively find this stage s. Recursively in A we can then see if the computation has been injured afterwards, i.e. if some element used negatively has entered A. If not, then $z_e = z_{e,1}$. Otherwise, $z_e = z_{e,0}$. $\square$

X.1.4 shows that the proof by the priority method given above has a nonrecursive element. Indeed, although we build A and B by a recursive construction that effectively chooses witnesses, there is no recursive criterion to tell whether a temporary witness is permanent or not.

We now prove that, in the terminology of XI.1.9, the set A constructed above is automatically low.

Corollary X.1.5 *$A \oplus B \equiv_T \mathcal{K}$ and $A' \equiv_T \mathcal{K}$.*

Proof. $A \oplus B \leq_T \mathcal{K}$ because A and B are r.e. Let h be a recursive function such that

$$\{h(x)\}^A(z) \simeq 0 \Leftrightarrow x \in \mathcal{K}.$$

Then

$$x \in \mathcal{K} \Leftrightarrow \{h(x)\}^A(z_{h(x)}) \simeq 0 \Leftrightarrow z_{h(x)} \in B.$$

Since z_e is recursive in $A \oplus B$, it follows that $\mathcal{K} \leq_T A \oplus B$.

$\mathcal{K} \leq_T A'$ always holds. Let g be a recursive function such that

$$\{g(x)\}^A(z) \simeq 0 \Leftrightarrow x \in A'.$$

Then

$$x \in A' \Leftrightarrow \{g(x)\}^A(z_{g(x)}) \simeq 0 \Leftrightarrow z_{g(x)} \in B$$

and thus $A' \leq_T A \oplus B \leq_T \mathcal{K}$. $\square$

Since $z_e = \lim_{s\to\infty} z_e^s$, from the Limit Lemma IV.1.17 we immediately have that z_e is recursive in $\mathcal{K}$. The estimate of X.1.4 is not better than this, because $A \oplus B \equiv_T \mathcal{K}$ by the corollary. But we had to use the former to prove the latter.

That $A \oplus B \equiv_T \mathcal{K}$ is quite amusing. We set out to build an incomplete r.e. set A, we needed an auxiliary set B, and thus the construction really built a single set $A \oplus B$, which turned out to be complete (this is another manifestation of the effectivity principle of p. I.280).

Exercises X.1.6 a) *Prove X.1.2 by satisfying the following requirements:*

$$R_e : A \not\simeq \{e\} \quad \textit{and} \quad N_e : B \not\simeq \{e\}^A.$$

(Hint: choose witnesses also for R_e.)

b) *Show that A and B are not simple.* (Hint: see X.1.3.)

The following improvement of the exercise above is the original solution to Post's Problem, and is the analogue of V.2.2.

Proposition X.1.7 Friedberg-Muchnik Theorem (Muchnik [1956], Friedberg [1957a]) *There are two incomparable r.e. degrees.*

Proof. Build r.e. sets A and B satisfying the following requirements:

$$\begin{aligned} N_e^B &: A \not\simeq \{e\}^B \\ N_e^A &: B \not\simeq \{e\}^A. \end{aligned}$$

The situation is now symmetric between A and B, and each requirement is purely negative for one set, but it can have a positive or a negative effect on the other set. Witnesses have to be chosen for each requirement, and the only small change in the proof is that when we choose a new witness, we make sure that it is not already restrained by some other condition (and we change it if it later becomes restrained by conditions of higher priority). Clearly, there are now two restraint functions $r^A(e, s)$ and $r^B(e, s)$, originated by N_e^A and N_e^B. Except for these variations, the proof is as in X.1.2. $\Box$

As in X.1.5 one can show that $A \oplus B \equiv_T \mathcal{K}$, and that both A and B are low. This is interesting because we will see later (X.6.26) that any two such sets cannot have g.l.b. in the (r.e.) degrees, and thus they automatically witness that the r.e. degrees do not form a lattice.

The following analogues of V.2.7 and V.2.9 do not require new ideas, and we leave their proofs as exercises.

Proposition X.1.8 (Muchnik [1958a]) *There exists a countable, recursively independent set of r.e. degrees.*

Proof. Build an r.e. set A whose components are recursively independent, see V.2.7. $\Box$

Corollary X.1.9 (Sacks [1963]) *Every countable partial ordering is embeddable in the r.e. degrees.*

Proof. As in V.2.9. $\Box$

As in V.2.10, one then has that *the one-quantifier theory of $\mathcal{R}$ is decidable.*

Sacks agreement method

We turn now to the second method of proof mentioned on p. 456. We want to build an r.e. set A such that $\mathcal{K} \not\leq_T A$. More generally, we want to learn how to **avoid upper cones**, by building an r.e. set A such that $C \not\leq_T A$, for any given r.e. nonrecursive set C.

The method used here is the analogue of the e-splitting method of V.2.12, although the parallel has to be taken lightly. The e-splitting method, when pushed to its limits, does give a decision procedure for the two-quantifier theory of $\mathcal{D}$ (Shore [1978] and Lerman [1983], see p. I.490). For the r.e. degrees, on the one hand the decidability of the two-quantifier theory is still an open question, and on the other hand additional ideas are needed to prove many known true two-quantifier sentences, for example the analogues of V.2.13 and V.2.16 (see X.6.1 and X.6.5).

The method of ensuring $C \not\simeq \{e\}^A$ is the usual one, namely to have a witness z_e such that $C(z_e) \not\simeq \{e\}^A(z_e)$. If C were not given, we could *choose* z_e and define $C(z_e)$ accordingly. But C is given beforehand, so even if at a certain stage s we pick up $z_e \notin C_s$ and wait for $t > s$ such that $\{e\}_t^{A_t}(z_e)$ converges, we cannot decide to leave z_e out of C, or to put it in. We thus have to *find* z_e. At any given stage s, there is an element z such that $C_s(z) \not\simeq \{e\}_s^{A_s}(z)$, for the simple reason that C_s is a total function and $\{e\}_s^{A_s}$ is not. We may thus define

$$l(e,s) \;=\; \max\{z : (\forall y < z)[C_s(y) \simeq \{e\}_s^{A_s}(y)]\}$$

(l is called the **length of agreement** between C and $\{e\}^A$ at stage s, and gives the smallest element on which they disagree). If we preserve the computation $\{e\}_s^{A_s}(l(e,s))$ when it converges and $C_s(l(e,s))$ is final, we ensure $C \not\simeq \{e\}^A$. But we have no control over C, so the C-side might change afterwards, and this strategy is not enough by itself.

The idea of the method is to preserve $\{e\}_s^{A_s}(x)$ not only for $x = l(e,s)$, but also for $x < l(e,s)$. In other words, we preserve the initial segment of agreement *and* the first point of disagreement. We thus set

$$r(e,s) \;=\; \max\{u(e,x,s) : x \leq l(e,s)\}.$$

Using the nonrecursiveness of C we can argue that if $\{e\}^A$ is total, then we will eventually preserve a convergent computation $\{e\}_s^{A_s}(x) \not\simeq C_s(x)$ for some x such that $C_s(x)$ is final, and thus ensure $\{e\}^A \not\simeq C$.

We argue by contradiction. Suppose $C \simeq \{e\}^A$ and N_e has highest priority. First note that $l(e,s)$ is increasing. This is because the $\{e\}^A$-side is never injured, since N_e has highest priority. So if $l(e,t) < l(e,s)$ for $t > s$, it must be so because the C-side has changed. But C is an r.e. set, and its approximation can change only once (from 0 to 1) for any given argument. Hence C_t is final for $l(e,t)$ and, since $l(e,t) < l(e,s)$,

$$\{e\}_s^{A_s}(l(e,t)) \simeq \{e\}_t^{A_t}(l(e,t)),$$

and the construction freezes $\{e\}_t^{A_t}(l(e,t))$ forever (unless $l(e,t)$ becomes even shorter afterwards). If we then consider the smallest value of $l(e,s)$, there is a permanent disagreement on it between C and $\{e\}^A$, contrary to the hypothesis. Thus $l(e,s)$ cannot drop down.

But if $l(e,s)$ can only become larger, then we can recursively compute C, as follows. Given x, we find s such that $l(e,s) > x$. Then $C_s = C(x)$, since

$$(\forall t \geq s)[C_t(x) \simeq \{e\}_t^{A_t}(x) \simeq \{e\}^A(x) \simeq C(x)],$$

where the first equality holds because $l(e,t) \geq l(e,s) > x$, the second because all computations up to $l(e,t)$ are preserved, and the third by hypothesis.

Proposition X.1.10 (Sacks [1963a]) *Given any nonrecursive r.e. set C, there is a nonrecursive r.e. set A such that $C \not\leq_T A$.*

Proof. The requirements are:

$$\begin{array}{rcl} P_e & : & \mathcal{W}_e \text{ infinite } \Rightarrow \mathcal{W}_e \cap A \neq \emptyset \\ N_e & : & C \not\simeq \{e\}^A. \end{array}$$

Let $\{C_s\}_{s\in\omega}$ be a recursive enumeration of C, and

$$\begin{array}{rcl} l(e,s) & = & \max\{z : (\forall y < z)[C_s(y) \simeq \{e\}_s^{A_s}(y)]\} \\ r(e,s) & = & \max\{u(e,x,s) : x \leq l(e,s)\}. \end{array}$$

The construction is as follows. Let $A_0 = \emptyset$. At stage $s+1$, look for the least $e \leq s$ such that $\mathcal{W}_{e,s} \cap A_s = \emptyset$ and, for some x,

$$x \in \mathcal{W}_{e,s} \ \wedge\ x \geq 2e \ \wedge\ (\forall i \leq e)[x > r(i,s)].$$

If such an e exists, put the least such x into A. Otherwise, do nothing.

- *N_e is satisfied, for every e*
  For the reasons explained in the motivations above, it is enough to wait for a stage s after which N_e is never injured. Such a stage exists because, as in X.1.2, each N_e can be injured at most finitely many times.

- $\lim_{s\to\infty} r(e,s) < \infty$, *for every e*
  Let $z_e = \mu z.\,(C(z) \not\simeq \{e\}^A(z))$, which exists because N_e is satisfied. Choose s_0 large enough so that, for every $s \geq s_0$:

  $$(\forall y < z_e)[\{e\}_s^{A_s}(y) \simeq \{e\}^A(y)]$$
  $$(\forall y \leq z_e)[C_s(y) = C(y)]$$
  $$N_e \text{ is not injured at stage } s.$$

  There are two cases. If $\{e\}_s^{A_s}(z_e)$ is never defined after s_0 then, by definition,
  $$\lim_{s\to\infty} r(e,s) = r(e,s_0).$$
  Otherwise, let s_1 be the first stage at which it is defined. then
  $$\lim_{s\to\infty} r(e,s) = r(e,s_1).$$

- *P_e is satisfied, for every e*
  As in X.1.2, using the previous fact. □

In X.1.2 it was possible to deduce the existence of z_e (which is the witness of $B \not\simeq \{e\}^A$) directly, since N_e was injured only finitely many times, and

z_e was the only witness active at any stage after which no injury occurred. Now the situation is quite different, and the existence of z_e has been deduced indirectly, by contradiction. Also, it is not true that the final witness is the apparent one at any stage after which no injury occurred, since the length of agreement between C and $\{e\}^A$ may still change many (although only finitely many) times, until it settles down to a final value.

Exercises X.1.11 a) *Given any nonrecursive Δ_2^0 set C, there is a nonrecursive r.e. set A such that $C \not\leq_T A$.* (Sacks [1963a]) (Hint: by the Limit Lemma, there is a recursive approximation $\{C_s\}_{s\in\omega}$ to C. Use, in place of l, the **modified length of agreement** function

$$m(e,s) \;=\; \max\,\{l(e,t) : t \leq s\},$$

where l is defined as before.)

b) *Given a uniformly r.e. sequence $\{C_n\}_{n\in\omega}$ of nonrecursive r.e. sets, there is a nonrecursive r.e. set A such that, for every n, $C_n \not\leq_T A$.* (Hint: enumerate the requirements, and proceed as in X.1.10).

c) *For any reducibility $\leq_r$ between $\leq_m$ and $\leq_T$, the r.e. r-degrees are not recursively enumerable without repetitions.* (Yates [1969], Kallibekov [1971]) (Hint: there are only finitely many degrees containing recursive sets, and by dropping their representatives we would still obtain a uniformly r.e. sequence of nonrecursive r.e. sets, representing all the nonrecursive r.e. degrees, contradicting part b).)

A tactical variation $\star$

To ensure $C \not\simeq \{e\}^A$ one can use the following alternative method, due to Yates [1966a].

By III.2.14, we may suppose that C is simple. It is enough to find $z \in C$ such that $\{e\}^A(z) \simeq 0$. We preserve computations of the form $\{e\}_s^{A_s}(x) \simeq 0$, for an r.e. set of x's. If it is only possible to do this for finitely many x, then we win because C is coinfinite and there is a $z \in \overline{C}$ such that $\{e\}^A(z) \not\simeq 0$. Otherwise, we still win because we now have an infinite r.e. set of x's, which must intersect C by the simplicity assumption.

Note that even with this approach, as already with the Sacks agreement strategy, we can only argue that a witness exists by an indirect argument, but not exhibit one directly.

X.2 Effective Baire Category $\star$

In this section we give an abstract formulation, due to Sacks [1963], of the type of finite injury arguments introduced in Section 1. We stress the analogies with the Baire category method of Section V.3. Different and stronger analogies have been proposed by Lachlan [1973] and Yates [1974].

Categorical formulation of finite injury arguments

We might say that the constructions of Section 1 amounted to a simultaneous definition of an r.e. set A and a recursive function t which indicated, for any s, which requirement we were trying to meet at stage $s+1$, and how (by specifying two finite sets, one to be put in A and the other to be kept out of it).

Definition X.2.1 *A* **requirement** *is a collection of ordered pairs of finite disjoint sets of natural numbers:*

$$R \;=\; \{(F_i, H_i) : i \in I\}.$$

The function t **enumerates requirements** *if, for every $s \geq 0$,*

$$D_{(t(s))_1} \cap D_{(t(s))_2} = \emptyset,$$

i.e. $t(s)$ codes a pair of disjoint finite sets.

In other words, a requirement is an open set, and t enumerates requirements if it enumerates basic open sets. When t enumerates requirements, we let

$$R_e \;=\; \{(F_s, H_s) : g(s) = e\},$$

where $F_s = D_{(t(s))_1}$, $H_s = D_{(t(s))_2}$ and $g(s) = (t(s))_2$.

Definition X.2.2 *R_e is* **met** *at stage $s+1$ if $e = g(s)$ and*

$$F_s \not\subseteq A_s \;\wedge\; F_s \subseteq A_{s+1} \;\wedge\; H_s \cap A_{s+1} = \emptyset.$$

R_e is **injured** *at stage $s+1$ if it was met at some previous stage $t+1$, and*

$$H_t \cap A_s = \emptyset \quad \textit{but} \quad H_t \cap A_{s+1} \neq \emptyset.$$

Note that if A_s is monotone in s (i.e. $A_s \subseteq A_{s+1}$), $A = \bigcup_{s\in\omega} A_s$, and R_e is met at some stage $s+1$ and not injured afterwards, then $F_s \subseteq A$ and $H_s \subseteq \overline{A}$, and A is in the open set

$$\mathcal{A}_e = \{X : (\exists s)[g(s) = e \;\wedge\; F_s \subseteq X \;\wedge\; H_s \subseteq \overline{X}]\}.$$

$\mathcal{A}_e$ is open because it is the union, via g, of the basic open sets determined by F_s and H_s. Moreover, if t is recursive, then $\mathcal{A}_e$ is effectively open.

Definition X.2.3 *To each recursive function t enumerating requirements we associate the r.e. set A (called the* **priority set** *of t) defined by stages as follows.*

We start with $A_0 = \emptyset$. At stage $s+1$, given A_s and $t(s)$, and hence F_s, H_s and $g(s)$, let $A_{s+1} = A_s$ (i.e. do not do anything) if one of the following holds:

1. *$H_s \cap A_s \neq \emptyset$, since we cannot meet $R_{g(s)}$ at stage $s+1$.*

2. *There is a stage $t < s$ such that $g(t) = g(s)$ and $R_{g(t)}$ was met at stage $t+1$ and not injured afterwards (i.e. $F_t \not\subseteq A_t$, $F_t \subseteq A_{t+1}$, and $H_t \cap A_s = \emptyset$), since we already have a chance of obtaining $A \in \mathcal{A}_{g(s)} = \mathcal{A}_{g(t)}$.*

3. *There is a stage $t < s$ such that $g(t) < g(s)$, $R_{g(t)}$ was met at stage $t+1$ and not injured afterwards, but it would be injured if we met $R_{g(s)}$ now (i.e. $F_t \not\subseteq A_t$, $F_t \subseteq A_{t+1}$, $H_t \cap A_s = \emptyset$ and $H_t \cap F_s \neq \emptyset$).*

In all other cases, let $A_{s+1} = A_s \cup F_s$. Note that, since case 1 does not hold, $H_s \cap A_s = \emptyset$. By hypothesis $H_s \cap F_s = \emptyset$, and so $H_s \cap A_{s+1} = \emptyset$. Hence if $F_s \not\subseteq A_s$, we meet $R_{g(s)}$ now.

Note that condition 3 above means that requirements with smaller indices have higher priority. Also, since the definition of A_{s+1} only involves the values of t up to s, it is not necessary to define t beforehand, but *t and A can be defined simultaneously*.

The crucial fact about the finite extension method in categorical form (V.3.13) was the possibility of meeting a countable collection of requirements R_n such that

$$(\forall \sigma)(\exists \tau \supseteq \sigma)(\forall A \supseteq \tau)(A \in R_n).$$

The latter property was shared by every open set (V.3.6), and amounted simply to the possibility of satisfying a given condition starting from any finite position. We now consider the analogous property.

Definition X.2.4 *The requirement R_e is* **dense** *if there is a chance of meeting it starting from any given position, i.e. for every finite set B there is a stage s such that $g(s) = e$ and*

1. *$H_s \cap A_s = \emptyset$, i.e. there is no obstacle to meeting R_e*

2. *$F_s \cap B = \emptyset$, i.e. R_e can be met avoiding B.*

The finite extension method produced a set in the intersection of a countable collection of open dense sets (V.3.13). The finite injury priority method produces an r.e. set in the intersection of an r.e. collection of effectively open dense sets, as the next result shows.

Proposition X.2.5 The Finite Injury Priority Method (Sacks [1963]) *Given a recursive function enumerating requirements, its priority set meets every dense requirement.*

Proof. First we note the following combinatorial facts:

- if a requirement is met, it can be met again only after it has been injured (X.2.3.2)
- each requirement is injured at most finitely many times (by X.2.3.3 a requirement can be injured only to meet some requirement of higher priority).

Let now R_e be dense. From the two facts just noted, there is a stage s_0 after which no requirement R_i with higher priority (i.e. $i < e$) is either injured or met. Let $B = \bigcup_{s \leq s_0}(F_s \cup H_s)$. By density, there is an s such that

$$g(s) = e \ \wedge \ H_s \cap A_s = \emptyset \ \wedge \ F_s \cap B = \emptyset.$$

Then $s > s_0$, because for $s \leq s_0$ we have $F_s \cap B \neq \emptyset$. By definition, X.2.3.1 does not occur (since $H_s \cap A_s = \emptyset$), and neither does X.2.3.3 (higher priority conditions are met for the last time, if ever, at some stage $t \leq s_0$, and so $H_t \cap F_s = \emptyset$ because $F_s \cap B = \emptyset$). Then either R_e has been met and not injured afterwards, or is met at stage $s+1$. In both cases, by the choice of s_0, R_e is not injured afterwards and hence is permanently met. $\Box$

As in Section V.3, we have a number of methodological consequences:

1. *requirements can be taken care of separately, by showing that each of them is dense*
2. *we can freely combine constructions known to be performable separately, as long as the global list of requirements remains recursively enumerable* (i.e. there is a recursive function enumerating the requirements in the sense of X.2.1).

These facts were also true for the finite extension method, but are more useful in this context because the global construction (i.e. making individual strategies for single requirements cohere) is now more complicated.

Exercise X.2.6 *R_e can be injured at most 2^e times.* (Hint: by induction, using the facts proved in X.2.5.)

The permitting method

The next result shows that the permitting method (III.3.16) can always be applied to the previous construction.

Proposition X.2.7 (Soare [1972]) *Given any nonrecursive r.e. set C and any recursive function enumerating requirements, there is an r.e. set $A \leq_T C$ that meets every dense requirement.*

Proof. Let f be a recursive one-one function with range C. As usual, we want to avoid putting something into A unless it is permitted. Since at stage $s+1$ the elements that may want to go into A are those in F_s, we do not let them go in unless $f(s) \leq \min F_s$. Except for this, the construction is as in X.2.5.

We only have to show that we still meet every dense requirement. Suppose R_e is the least requirement that is not satisfied, and let s_0 be a stage after which R_e has highest priority. If R_e is met at some stage after s_0, then it is never injured and is permanently satisfied. Then, by hypothesis, R_e is never met. But by density we do have infinitely many chances to meet it. Since we never do it, it must be that at any stage s in which we have a chance, $f(s) \not\leq \min F_s$. We can thus define an increasing sequence of stages $s_1, s_2, \dots$ such that $\min F_{s_n}$ is increasing and not permitted. But then C is recursive, since to see if $x \in C$, we only have to search for n large enough so that $x < \min F_{s_n}$. Then $x \in C$ if and only if x has been generated in C before stage s_n. □

Corollary X.2.8 (Muchnik [1956], [1958a], Sacks [1963])

1. *There is no minimal r.e. degree.*
2. *Every countable partial ordering is embeddable in the r.e. degrees below any given nonzero r.e. degree.*
3. *Every nonrecursive r.e. degree bounds a nonrecursive low degree.*

Proof. From the existence results of Section 1. □

Notice that the same results also hold for the r.e. *wtt*-degrees, because permitting preserves *wtt*-reducibility (p. I.338).

The strong effective version of the Baire category method just proved has another interesting methodological consequence, namely that *we cannot use the finite injury priority method in the form given above to produce an r.e. degree that is not low.* Otherwise, by the result just proved we could push the construction below any given nonrecursive r.e. degree, in particular below a nonrecursive low one; but every degree below a low one is low.

In particular, *we cannot use the finite injury priority method in the form given above to produce a maximal or a hyperhypersimple set* because, by IX.2.24 and IX.2.25.b, such sets are not low. Lachlan [1967a] has a more general formulation of the priority method, which also subsumes the construction of maximal sets.

Degrees r.e. in smaller degrees ⋆

The relativization of X.1.2 produces a great number of degrees which are solutions to Post's Problem relative to some smaller degree. More precisely, for any degree $\boldsymbol{b}$ there is a degree $\boldsymbol{a}$ r.e. in it and strictly between $\boldsymbol{b}$ and $\boldsymbol{b'}$. Not every degree can be the solution to Post's Problem relative to some smaller degree (for example, a minimal degree is not, since the only degree below it is $\mathbf{0}$, and a minimal degree is not r.e. by X.2.8.1). Thus, in a sense, the next result is the best possible one.

Proposition X.2.9 (Jockusch [1981]) *The set of degrees r.e. in some smaller degree is comeager.*

Proof. It is enough to build, by finite extensions, a set A which determines a set B such that A is r.e. in B and $B <_T A$. We view A and B as subsets of $N \cdot N$ and notice that if

$$\langle x, y\rangle \in B \iff x \in A \wedge \langle x, y\rangle \notin A,$$

then $B \leq_T A$ by definition. If A is immune, then it is r.e. in B, since then

$$x \in A \iff (\exists y)(\langle x, y\rangle \in B).$$

The right-to-left direction holds by definition. The left-to-right direction holds because, for any x, the set $\{\langle x, y\rangle : y \in \omega\}$ is r.e. and infinite, and if A is immune some element of it is not in A.

Since the set of immune sets is comeager (V.5.3.c) and so is the intersection of two comeager sets, it is enough to build by finite extensions a set A such that $A \not\leq_T B$. As usual, it is enough to show how to deal with a single requirement $A \not\simeq \{e\}^B$ at stage s.

Given τ, let τ' be the unique string such that $|\tau| = |\tau'|$ and

$$\tau'(\langle x, y\rangle) = \begin{cases} 1 & \text{if } \tau(x) = 1 \wedge \tau(\langle x, y\rangle) = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Given σ_s, fix z outside its domain and not of the form $\langle x, y\rangle$, and see if there is $\tau \supseteq \sigma_s$ such that $\{e\}^{\tau'}(z) \simeq 0$.

If not, we let σ_{s+1} be any extension of σ_s that gives the value 0 to z (so that $A(z) = 0$ but $\{e\}^B(z) \not\simeq 0$).

Otherwise, pick one such τ. We want to show that there is a string $\mu \supseteq \tau$ such that $\mu' \supseteq \tau'$ and $\mu(z) = 1$. Then we can let $\sigma_{s+1} = \mu$ (so that $A(z) = 1$ but $\{e\}^B(z) \simeq 0$).

The only nontrivial case is when $\tau(z) = 0$. To define μ, let it agree with τ on the numbers on which τ is 1 and let it be 0 on z. Note that this does not change the effect of τ on B, since z is not of the form $\langle x, y\rangle$. Whenever we let μ be 1 for some n on which τ was 0 (e.g. for z), we want to avoid any change in the effect on B, i.e. we want μ' to be 0 for the elements $\langle n, y\rangle$ on which τ' was 0. But since now $\mu(n) = 1$, it is enough to let $\mu(\langle n, y\rangle) = 1$, so that we still have $\mu'(\langle x, y\rangle) = 0$. $\square$

Corollary X.2.10 *The set of degrees that are solutions to Post's Problem relative to some smaller degree is comeager.*

Proof. The result just proved implies that the set

$$\mathcal{A} = \{\boldsymbol{a} : (\exists \boldsymbol{b})(\boldsymbol{b} < \boldsymbol{a} \ \wedge \ \boldsymbol{a} \text{ r.e. in } \boldsymbol{b})\}$$

is comeager. Note that if $\boldsymbol{a}$ is r.e. in $\boldsymbol{b}$, then $\boldsymbol{a} \leq \boldsymbol{b}'$. But the set

$$\mathcal{B} = \{\boldsymbol{a} : (\exists \boldsymbol{b})(\boldsymbol{a} = \boldsymbol{b}')\}$$

is meager, since all of its members are comparable with $\mathbf{0}'$ (V.3.14). Thus

$$\mathcal{A} \cap \overline{\mathcal{B}} = \{\boldsymbol{a} : (\exists \boldsymbol{b})(\boldsymbol{b} < \boldsymbol{a} < \boldsymbol{b}' \ \wedge \ \boldsymbol{a} \text{ r.e. in } \boldsymbol{b})\}$$

is comeager. □

Kurtz [1981] has proved that the set of degrees r.e. in some smaller degree also has measure 1.

X.3 The Infinite Injury Priority Method

We have seen in the last section that methods more powerful than finite injury X.2.5 are needed, if we wish to prove results such as the existence of r.e. degrees that are not low. This is not surprising, since after all the finite injury priority method is just a constructive version of the finite extension method (the fact that there are only finitely many injuries comes precisely from the fact that a requirement can be satisfied by finitary means).

We now introduce a constructive version of the coinfinite extension method of Section V.4. However, we will later see that for many applications to strong reducibilities the finite injury priority method is sufficient. More sophistication is necessary only for the study of r.e. T-degrees and for complicated results about r.e. *wtt*-degrees.

Thick subsets

A typical case in which we have to injure a computation infinitely many times is when we are given an r.e. nonrecursive set B, and want to build an r.e. set A such that $B \leq_T A$. To ensure this, we code B into the 0-th column of A, by letting

$$\langle 0, z\rangle \in A \ \Leftrightarrow \ z \in B$$

with highest priority.

Another typical, more general case is when we want to build an upper bound of an r.e. sequence of r.e. sets $\{B_e\}_{e\in\omega}$ (satisfying some additional conditions), and in this case we ensure that, for each e,

$$\langle e, z\rangle \in A \ \Leftrightarrow\ z \in B_e,$$

with at most finitely many exceptions.

The following concepts were used implicitly in the proof of V.4.3.

Definition X.3.1 *Given a set Z, we let*

1. $\boldsymbol{Z^{[e]}} = \{\langle e, z\rangle : \langle e, z\rangle \in Z\}$ *(the e-th* **column** *of Z)*
2. $\boldsymbol{Z^{[<e]}} = \bigcup_{n<e} Z^{[n]}$ *(the e-th* **section** *of Z).*

A is a **thick subset** *of B if $A \subseteq B$ and their columns differ at most finitely, i.e. $A^{[e]} =^* B^{[e]}$ for every e.*

As a first (and, as we will see, quite typical) application of the infinite injury priority method, given an r.e. nonrecursive set C and an r.e. set B we want to build a thick subset A of B such that $C \not\leq_T A$. The requirements are:

$$\begin{aligned} P_e &\ :\ B^{[e]} \subseteq^* A \\ N_e &\ :\ C \not\simeq \{e\}^A. \end{aligned}$$

Note that, since we will have $A \subseteq B$, $A^{[e]} \subseteq B^{[e]}$ will automatically be ensured, and thus P_e implies $A^{[e]} =^* B^{[e]}$.

Abstracting from this particular problem a little further, we have to deal with positive requirements of the form

$$P_e \ :\ \mathcal{W}_{f(e)} \subseteq^* A,$$

where f is recursive and $\mathcal{W}_{f(e)}$ is a possibly infinite r.e. set.

The Injury Lemma

We follow Sacks [1964b] in analyzing the requirements one at a time. Suppose we fix the following usual priority ordering:

$$N_0 > P_0 > N_1 > P_1 > \cdots$$

and examine the steps necessary to satisfy them in this order.

The construction is like those in Section 1, and it consists of putting elements into A to satisfy the positive requirements, as long as these elements are not restrained by negative requirements of higher priority.

We will define recursive functions

$$\begin{aligned}\hat{l}(e,s) &= \text{length of agreement function}\\ \hat{r}(e,s) &= \text{restraint function}\end{aligned}$$

and sets (not necessarily finite, or even recursive)

$$\hat{I}_e \quad = \quad \{x : x \in A_{s+1} - A_s \ \wedge \ (\exists s)[x \leq \hat{r}(e,s)]\},$$

different from but with the same interpretation as l, r and I_e in Section 1. We can thus use them immediately, even if their formal definitions are deferred for a little while (see X.3.4).

N_0 has highest priority and is never injured. It can thus be satisfied by the Sacks agreement method of Section 1. Then the restraint function is bounded (i.e. $\lim_{s\to\infty} \hat{r}(0,s) < \infty$), and this allows the satisfaction of P_0. Now P_0 is possibly infinitary, and it may injure N_1 infinitely many times, thus preventing its satisfaction. However, a simple look at the proofs of Section 1 shows that it is not necessary to have only finitely many injuries to be able to satisfy a negative requirement N_e. It is enough to be able to control the elements that injure N_e, e.g. to have $\hat{I}_e$ recursive. Actually, an even weaker condition suffices.

Proposition X.3.2 Injury Lemma (Sacks [1964b], Soare [1976]) *If $C \not\leq_T \hat{I}_e$, then $C \not\simeq \{e\}^A$, i.e. N_e is satisfied.*

Proof. Suppose $C \simeq \{e\}^A$. Then $\lim_{s\to\infty} \hat{l}(e,s) = \infty$. Given x, find s such that

$$\hat{l}(e,s) > x$$
$$(\forall z \leq x)(\{e\}_s^{A_s}(z) \text{ has the final value}),$$

i.e.

$$(\forall z \leq x)(\forall y)[y \leq u(e,z,s) \ \Rightarrow \ y \notin \hat{I}_e \ \vee \ y \in A_s].$$

The search for s is clearly recursive in $\hat{I}_e$, and s exists because $\{e\}^A$ is total. Then, for all $t \geq s$,

$$\hat{l}(e,t) > x$$
$$\hat{r}(e,t) > \max\{u(e,z,s) : z \leq x\},$$

and thus

$$C_s(x) \simeq \{e\}_s^{A_s}(x) \simeq \{e\}^A(x) \simeq C(x).$$

Then $C \leq_T \hat{I}_e$, which is a contradiction. $\square$

For strong reducibilities, only an analogue of the Injury Lemma is needed to handle infinitary positive requirements. Indeed, if N_e is satisfied, then we can proceed exactly as in X.1.10, by letting $z_e = \mu z. [C(z) \not\simeq \{e\}^A(z)]$ and choosing s_0 large enough so that, for every $s \geq s_0$,

$$(\forall y \leq z_e)[\{e\}_s^{A_s}(y) \simeq \{e\}^A(y)]$$
$$(\forall y \leq z_e)[C_s(y) = C(y)].$$

Then $\lim_{s\to\infty} r(e,s) = r(e,s_0)$. The crucial fact here is that even if $\{e\}^A(z_e)$ does not converge, then for all sufficiently large s, $\{e\}_s^{A_s}(z_e)$ does not converge either (because $\{e\}^A(z_e)$ queries the oracle only up to elements bounded by a recursive function, and when A has settled on those elements the computation is final). Thus in this case we can use the old l and r as $\hat{l}$ and $\hat{r}$.

For Turing reducibility, we can only find s_0 large enough so that

$$(\forall y < z_e)[\{e\}_s^{A_s}(y) \simeq \{e\}^A(y)]$$

(because these computations converge) and

$$(\forall y \leq z_e)[C_s(y) = C(y)].$$

In X.1.10 we were able to override the difficult case of when $\{e\}^A(z_e)$ does not converge by going to a stage after which N_e is no longer injured (since such a stage has the property that if $\{e\}_s^{A_s}(z_e)$ converges after it, then we preserve the computation forever). But this required finitary positive requirements.

The Hat Trick

We now know how to satisfy N_1 any time the positive requirement P_0 is sufficiently simple to produce a tame injury set $\hat{I}_1$. This will have to be verified in each particular construction. Then N_1 is satisfied under the hypothesis $C \not\leq_T \hat{I}_1$. But since it might have been injured infinitely many times by P_0, we cannot deduce that $\lim_{s\to\infty} \hat{r}(1,s) < \infty$. Indeed, in the notation at the end of the previous subsection, for $s \geq s_0$,

$$\hat{l}(1,s) \geq z_1$$
$$\hat{r}(1,s) \geq u(1,z_1,s)$$

but if $\{1\}^A(z_1)$ does not converge, it might be that $\lim_{s\to\infty} u(1,z_1,s) = \infty$.

The reason why this can happen is that a computation $\{e\}^A(x)$ can diverge even if the intermediate computations $\{e\}_s^{A_s}(x)$ converge infinitely often. This obviously implies that $\lim_{s\to\infty} u(e,x,s) = \infty$. Otherwise, after A has settled on all the elements less than the limit, the computation would be final.

Proposition X.3.3 *If A is r.e. and, for every e and x,*

$$\{e\}_s^{A_s}(x)\!\downarrow \text{ for infinitely many } s \;\Rightarrow\; \{e\}^A(x)\!\downarrow,$$

then $A' \leq_T \mathcal{K}$, i.e. A is low.

Proof. Let
$$g(e,s) = \begin{cases} 1 & \text{if } \{e\}_s^{A_s}(e)\!\downarrow \\ 0 & \text{otherwise.} \end{cases}$$

Then g is recursive and

$$e \in A' \;\Leftrightarrow\; \lim_{s\to\infty} g(e,s) = 1,$$

since:

- if $e \in A'$, then $\{e\}^A(e)\!\downarrow$ and $\lim_{s\to\infty} g(e,s) = 1$
- if $e \notin A'$, then $\{e\}^A(e)\!\uparrow$ and by hypothesis $\{e\}_s^{A_s}(e)\!\downarrow$ only for finitely many s. So $\lim_{s\to\infty} g(e,s) = 0$.

Then, by the Limit Lemma, $A' \leq_T \mathcal{K}$. $\square$

Thus if A is not low (e.g. if it is $\mathcal{K}$ itself) there are e and infinitely many x such that $\lim_{s\to\infty} u(e,x,s) = \infty$.

A moment of reflection, however, tells us that to satisfy the positive requirement P_1, we do not need to have $\lim_{s\to\infty} \hat{r}(1,s) < \infty$. This would only *simplify* things, since then, from a certain point on, N_1 would no longer interfere (exactly as N_0 did). But we really just need infinitely many chances, i.e.

$$\liminf_{s\to\infty} \hat{r}(1,s) < \infty,$$

because then every element that we want to put into A and is larger than the lim inf will indeed get into A at an appropriate stage (when $\hat{r}(1,s)$ drops back to its lim inf).

Since the only reason why $\lim_{s\to\infty} \hat{r}(1,s)$ could go to infinity is because a computation is injured infinitely often, we decide to modify the computations in such a way that, when we have an injury, the restraint function drops back to 0. This will cause no harm, since:

- either a computation is not injured after a certain stage, and then the modified computation will agree with it from then on
- or the computation is injured infinitely many times, and then we can certainly modify it between two consecutive injuries with no trouble.

The idea is thus to let

$$\{\hat{e}\}_s^{A_s}(x) \simeq \begin{cases} \{e\}_s^{A_s}(x) & \text{if defined and not injured at step } s \\ \text{undefined} & \text{otherwise,} \end{cases}$$

where the 'otherwise' case means that either $\{e\}_s^{A_s}(x)$ is undefined, or it converges with a computation different from $\{e\}_{s-1}^{A_{s-1}}(x)$.

Since a computation is injured at step s if something less than or equal to the use function enters A at that stage, we let

$$a_s = \begin{cases} \mu x.\,(x \in A_s - A_{s-1}) & \text{if } A_s - A_{s-1} \neq \emptyset \\ a_{s-1} & \text{otherwise.} \end{cases}$$

This makes $\lim_{s\to\infty} a_s = \infty$ whenever A is infinite, which is the only case of interest in the applications.

Definition X.3.4 The Hat Trick (Soare [1976])

$$\begin{aligned} \{\hat{e}\}_s^{A_s}(x) &\simeq \begin{cases} \{e\}_s^{A_s}(x) & \textit{if defined and } u(e,x,s) < a_s \\ \textit{undefined} & \textit{otherwise} \end{cases} \\ \hat{u}(e,x,s) &= \begin{cases} u(e,x,s) & \textit{if } \{\hat{e}\}_s^{A_s}(x)\downarrow \\ 0 & \textit{otherwise} \end{cases} \\ \hat{l}(e,s) &= \max\,\{z : (\forall y < z)[C_s(y) \simeq \{\hat{e}\}_s^{A_s}(y)]\} \\ \hat{r}(e,s) &= \max\,\{\hat{u}(e,x,s) : x \leq \hat{l}(e,s)\}. \end{aligned}$$

Now $\lim_{s\to\infty} \hat{r}(0,s) < \infty$ and $\liminf_{s\to\infty} \hat{r}(1,s) < \infty$, so P_1 can be satisfied. To satisfy N_2, the Injury Lemma will suffice as long as the hypotheses on $\hat{I}_2$ are satisfied.

The Window Lemma

We now turn to P_2, and another (final) problem arises. We now have

$$\begin{gathered} \lim_{s\to\infty} \hat{r}(0,s) < \infty \\ \liminf_{s\to\infty} \hat{r}(1,s) < \infty \\ \liminf_{s\to\infty} \hat{r}(2,s) < \infty \end{gathered}$$

but it could be that $\hat{r}(1,s)$ and $\hat{r}(2,s)$ do not drop back simultaneously so that, if

$$\hat{R}(e,s) \;=\; \max_{i\leq e}\; \hat{r}(i,s),$$

then

$$\liminf_{s\to\infty} \hat{R}(2,s) = \infty.$$

But this is not the case, as we now see by considering the nondeficiency or true stages (II.6.16) of the enumeration of A.

Let

$$s \in T \ \Leftrightarrow\ (\forall t \geq s)(a_t \geq a_s).$$

From II.6.16 we know that T is infinite, and the crucial fact is the following:

if $s \in T$, then computations convergent at stage s are permanent, i.e.

$$\{\hat{e}\}_s^{A_s}(x) \simeq y \ \Rightarrow\ \{e\}^A(x) \simeq y,$$

because if $\{\hat{e}\}_s^{A_s}(x)\downarrow$, then $u(e,x,s) < a_s$. Indeed, if $s \in T$, then $a_t \geq a_s$ and $\{\hat{e}\}_t^{A_t}(x) \simeq \{\hat{e}\}_s^{A}(x)$ for every $t \geq s$.

Proposition X.3.5 Window Lemma (Robinson [1969], Lachlan [1973], Soare [1976])

1. *If $C \not\simeq \{e\}^A$, then* $\lim_{t\in T} \hat{r}(e,t) < \infty$.
2. *If $(\forall i \leq e)(C \not\simeq \{i\}^A)$, then* $\liminf_{s\to\infty} \hat{R}(e,s) < \infty$.

Proof. If $C \not\simeq \{e\}^A$, let $z_e = \mu z.\,[C(z) \not\simeq \{e\}^A(z)]$. Choose s_0 large enough so that, for every $s \geq s_0$,

$$(\forall y < z_e)[\{\hat{e}\}_s^{A_s}(y) \simeq \{e\}^A(y)]$$
$$(\forall y \leq z_e)[C_s(y) = C(y)].$$

There are two cases:

- if $\{\hat{e}\}_s^{A_s}(z_e)$ is never defined for $s \in T$ such that $s \geq s_0$, then

$$\lim_{t\in T} \hat{r}(e,t) = r(e,s_0)$$

- if $\{\hat{e}\}_s^{A_s}(z_e)$ is defined for some $s \in T$ such that $s \geq s_0$, let s_1 be the smallest such s. Then

$$\lim_{t\in T} \hat{r}(e,t) = r(e,s_1).$$

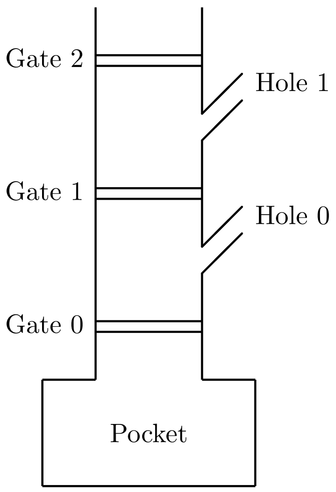
This proves part 1. Part 2 follows from it. □

The Injury and Window Lemmas are very general and allow us to satisfy all the requirements, as long as we can prove that the positive ones are sufficiently well-behaved to produce injury sets $\hat{I}_e$ satisfying the required hypotheses.

The Gate Opening Lemma $\star$

The use of nondeficiency stages to prove the Window Lemma is very elegant, but it looks very much as a trick or a miracle. It is thus useful to pause for a moment and consider an alternative way of solving the problem it addresses.

The main idea is the following. Given an element x that might go into A for the sake of a positive requirement, in the previous setting we waited for a stage at which x was permitted to go in by all negative requirements of higher priority *simultaneously*. Now instead we allow for these negative requirements to permit one at a time, and put x into A when it has been permitted by all of them *successively*.



The best pictorial explanation of what goes on in the construction has been proposed by Lerman [1973], in terms of pinball machines. The numbers involved in the construction correspond to *balls* that the set B ejects through *holes*, one for each positive requirement. The balls roll down towards a *pocket* corresponding to the set A, by passing through *gates* set up by the negative requirements, that open when the associated restraints drop back. The ordering of holes and gates reflects the priority ordering, as in the picture above.

We say that the e-th gate is open for x at stage $s+1$ if $x > \hat{r}(e,s)$, and it is closed otherwise. When an element is ejected by a given hole at a given stage, it rolls down through the machine, stops in front of the first gate that is closed for it and waits for the gate to open for it at a later stage, in which case it starts rolling down again. Eventually the element reaches either a gate in front of which it waits permanently, or the pocket.

We let $G_{e,s}$ be the set of elements that are waiting in front of the e-th gate at the end of stage s, and let G_e be the set of elements that permanently wait in front of it, i.e. for which the e-th gate never opens. We also let

$$G_{<e,s} = \bigcup_{i<e} G_{i,s} \qquad \text{and} \qquad G_{<e} = \bigcup_{i<e} G_i.$$

The essence of the pinball machine method is contained in the next result. Its two parts provide, respectively, analogues of the Injury and Window Lemmas.

Proposition X.3.6 Gate Opening Lemma (Shoenfield [1961], Sacks [1963b], Lerman [1973])

1. *If $C \not\leq_T A^{[<e]}$ and $G_{<e}$ is finite, then $C \not\simeq \{e\}^A$, i.e. N_e is satisfied.*
2. *If $C \not\simeq \{e\}^A$, then $\liminf_{s\to\infty} \hat{r}(e,s) < \infty$, i.e. G_e is finite.*

Proof. Part 1 is proved as in the Injury Lemma X.3.2. Suppose $C \simeq \{e\}^A$. Then $\lim_{s\to\infty} \hat{l}(e,s) = \infty$. Given x, find s such that

$$\hat{l}(e,s) > x$$
$$(\forall z \leq x)(\{e\}_s^{A_s}(z) \text{ has the final value}),$$

i.e.

$$(\forall z \leq x)(\forall y)[y \leq u(e,z,s) \ \wedge\ y \in A^{[<e]} \cup G_{<e} \ \Rightarrow\ y \in A_s^{[<e]} \cup G_{<e,s}].$$

The search for s is clearly recursive in $A^{[<e]}$, since $G_{<e}$ is finite. Moreover, s exists because $\{e\}^A$ is total. Then, for all $t \geq s$,

$$\hat{l}(e,t) > x$$
$$\hat{r}(e,t) > \max\{u(e,z,s) : z \leq x\},$$

and any element that enters A at a stage $t \geq s$ must be greater than $\hat{r}(e,t)$, since it has to go through the e-th gate. Thus

$$C_s(x) \simeq \{e\}_s^{A_s}(x) \simeq \{e\}^A(x) \simeq C(x)$$

and $C \leq_T A^{[<e]}$, contradiction.

Part 2 is proved as in the Window Lemma X.3.5. If $C \not\simeq \{e\}^A$, then let $z_e = \mu z.\,[C(z) \not\simeq \{e\}^A(z)]$. Choose s_0 large enough so that, for every $s \geq s_0$:

$$(\forall y < z_e)[\{\hat{e}\}_s^{A_s}(y) \simeq \{e\}^A(y)]$$
$$(\forall y \leq z_e)[C_s(y) = C(y)].$$

There are two cases:

- if $\{e\}^A(z_e)$ is not defined, then there are infinitely many stages s such that $\{\hat{e}\}_s^{A_s}(z_e)$ is not defined, and
$$\liminf_{s\to\infty} \hat{r}(e,s) = r(e,s_0)$$
- if $\{e\}^A(z_e)$ is defined, then there is a stage $s_1 \geq s_0$ after which $\{\hat{e}\}_s^{A_s}(z_e)$ is always defined, and
$$\liminf_{s\to\infty} \hat{r}(e,s) = r(e,s_1). \quad \Box$$

The Gate Opening Lemma is more general than the Injury and Window Lemmas and allow us to satisfy all the requirements, as long as we can prove that the positive ones are sufficiently well-behaved to produce columns satisfying the needed hypotheses.

The Thickness Lemma

We now prove what has been, historically, the first application of the infinite injury priority method. We give two proofs, the first by the Injury and Window Lemmas and the second by the Gate Opening Lemma. A third proof, by the tree method, will be given in X.4.3.

Theorem X.3.7 Thickness Lemma (Shoenfield [1961], [1971a]) *Let C be an r.e. nonrecursive set, and B an r.e. set such that $C \not\leq_T B^{[<e]}$, for every e. Then there is a thick subset A of B such that $C \not\leq_T A$.*

First Proof. The requirements are:

$$\begin{aligned} P_e &: B^{[e]} \subseteq^* A \\ N_e &: C \not\simeq \{e\}^A. \end{aligned}$$

If we want to use the Injury and Window Lemmas, the construction is simply

$$x \in A_{s+1} \Leftrightarrow x \in B_{s+1}^{[e]} \wedge (\forall i \leq e)[x > \hat{r}(i,s)],$$

i.e. we put an element in A whenever it is in B, and it is not restrained by negative requirements of higher priority.

Note that:

- $\hat{I}_e \subseteq A^{[<e]}$
  x can injure N_e only if it is put in A to satisfy a positive requirement of higher priority, i.e. for P_i with $i < e$.

- $\hat{I}_e \leq_T A^{[<e]}$
  Given x, to know if it is in $\hat{I}_e$ first we see if it is in $A^{[<e]}$ (since $\hat{I}_e \subseteq A^{[<e]}$). If so, we find a stage s such that $x \in A^{[<e]}_s$. Then $x \in \hat{I}_e$ if and only if $x \in \hat{I}_{e,s}$.

We can now prove by induction that the requirements are satisfied. Suppose that P_i and N_i are satisfied for $i < e$.

- *N_e is satisfied*
  Since $B^{[i]} \subseteq^* A$ for every $i < e$, $B^{[<e]} =^* A^{[<e]}$. Then $C \not\leq_T \hat{I}_e$, otherwise $C \leq_T \hat{I}_e \leq_T A^{[<e]}$ and hence $C \leq_T B^{[<e]}$, contradicting the hypothesis. By the Injury Lemma, it follows that $C \not\simeq \{e\}^A$.

- *P_e is satisfied*
  Since $C \not\simeq \{i\}^A$ for every $i \leq e$, $\liminf_{s\to\infty} \hat{R}(e,s) < \infty$ by the Window Lemma and so, by construction, $B^{[e]} \subseteq^* A$. $\Box$

Second Proof. If we want to use the Gate Opening Lemma, the construction is as follows. At stage $s+1$, for each e and each $x \in B^{[e]}_{s+1} - B^{[e]}_s$ we eject x from the e-th hole, and place it in front of the e-th gate. Moreover, for each e and each x that is in front of the e-th gate, we place x in A_{s+1} if all gates of higher priority are open for it, i.e. $x > \hat{r}(i,s)$ for all $i \leq e$, and in front of the first gate of higher priority that is closed for it, otherwise.

We can now prove by induction that the requirements are satisfied. Suppose that P_i and N_i are satisfied for $i < e$.

- *N_e is satisfied*
  Since $B^{[i]} \subseteq^* A$ for every $i < e$, $B^{[<e]} =^* A^{[<e]}$. Then $C \not\leq_T A^{[<e]}$, otherwise $C \leq_T A^{[<e]} =^* B^{[<e]}$ and hence $C \leq_T B^{[<e]}$, contradicting the hypothesis. Since $C \not\simeq \{i\}^A$ for every $i < e$, G_i is finite by part 2 of the Gate Opening Lemma, and hence $G_{<e}$ is finite. By part 1 of the Gate Opening Lemma, it follows that $C \not\simeq \{e\}^A$.

- *P_e is satisfied*
  Since $C \not\simeq \{i\}^A$ for every $i \leq e$, $\liminf_{s\to\infty} \hat{r}(i,s) < \infty$ by part 2 of the Gate Opening Lemma and so, by construction, $B^{[e]} \subseteq^* A$. $\Box$

Corollary X.3.8 Localized Thickness Lemma. *If $C \not\leq_T B^{[<e]}$, then for every $i \leq e$, $B^{[i]} =^* A^{[i]}$ and $C \not\simeq \{i\}^A$.*

Shoenfield [1961] originally obtained the Thickness Lemma in the special case when B is *piecewise recursive* (i.e. $B^{[e]}$ is recursive for every e) and $C = \mathcal{K}$. The treatment of the infinite injury machinery in this case is not simpler than in the general case above. Piecewise recursiveness is clearly not needed in the

Injury Lemma. The reason why Shoenfield did not obtain the general case is that he did not know at the time how to handle the condition $C \not\leq_T A$ (this was Sacks' achievement in [1963a]). He thus obtained $\mathcal{K} \not\leq_T A$ by the first method used in Section 1, namely by simultaneously constructing an r.e. set nonrecursive in A.

The infinite injury priority method was later rediscovered independently by Sacks [1963b], and widely applied since. Then Shoenfield [1971a] had the last word, and showed how many results follow from the Thickness Lemma, either in the form above or in the stronger form below (see X.9.15 and its consequences).

Theorem X.3.9 Strong Thickness Lemma (Shoenfield [1971a]) *Let C be an r.e. nonrecursive set, and let B be an r.e. set such that $C \not\leq_T B^{[<e]}$, for every e. Then there is a thick subset A of B such that $C \not\leq_T A$ and $A \leq_T B$.*

Proof. The only embellishment here is the condition that $A \leq_T B$. To obtain it, we need to be able to recover A, i.e. the construction, from B. The problem we have is that the restraint function depends on the length of agreement function, and this in turn also depends on C (since it can be changed by changes on the C-side). We thus use a modified length of agreement function:

$$\begin{aligned} \hat{m}(e,s) &= \max\{z : z \leq \max_{t \leq s} \hat{l}(e,t) \ \wedge \ (\forall y < z)(\{\hat{e}\}_s^{A_s}(y)\downarrow)\} \\ \hat{r}(e,s) &= \max\{\hat{u}(e,x,s) : x \leq \hat{m}(e,s)\}. \end{aligned}$$

Intuitively, we continue to preserve all the computations we asked to preserve at some previous stage, even if the original length of agreement drops back, as long as they have not been injured. Thus the changes depend only on A, and hence on B.

The Injury and Window Lemmas are proved as before, and nondeficiency stages s now have the property that

$$(\forall t \geq s)[\hat{m}(e,s) \leq \hat{m}(e,t) \ \wedge \ \hat{r}(e,s) \leq \hat{r}(e,t)],$$

because computations are not injured afterwards.

To show that $A \leq_T B$, we prove that $A^{[e]} \leq_T B$ uniformly in e, by induction on e. Suppose $A^{[<e]} \leq_T B$. Given x, first see if $x \in B^{[e]}$. If so, find a stage s such that $x \in B_s^{[e]}$. At any stage $t \geq s$, x will go into $A_{t+1}^{[e]}$ unless it is restrained, i.e. if $x \leq \hat{r}(i,t)$ for some $i \leq e$. But recursively in $A^{[<e]}$ (and hence in B, by the inductive hypothesis), we can find a nondeficiency stage $t \geq s$ for $A^{[<e]}$, and we know that from this point on the restraint function will not drop back below $\hat{r}(i,t)$, for any $i \leq e$. So either $x \in A_{t+1}^{[e]}$ or $x \notin A^{[e]}$ (because $x \leq \max_{i \leq e} \hat{r}(i,t)$, and it will be so forever). $\Box$

Formal systems and r.e. sets ⋆

In III.10.4 we have discussed two direct methods of proving undecidability of a formal system, namely by weakly representing in it either every recursive set or some nonrecursive r.e. set, but have left open some questions about them.

In particular, we did not determine the consequences of weak representability of all recursive sets. For example, from a definability point of view, whether it implies the weak representability of some nonrecursive r.e. set, too (in which case the first method would be a special case of the second). Or, from a complexity point of view, whether it implies that the formal system is undecidable in the strongest possible sense, i.e. that the set of its theorems is creative, or at least T-complete.

A negative answer to both questions is given by the next result, whose proof has been the historical motivation for, as well as the first application of, the infinite injury priority method.

Proposition X.3.10 (Shoenfield [1961]) *There is a consistent formal system $\mathcal{F}$ whose weakly representable sets are exactly the recursive ones. Moreover, $\mathcal{F}$ can be chosen to be of degree less than $\mathbf{0}'$.*

Proof. Let $\{\mathcal{W}_{f(e)}\}_{e\in\omega}$ be a recursive enumeration of the recursive sets (see II.5.26), and $B = \bigoplus_{e\in\omega} \mathcal{W}_{f(e)}$: B is r.e. and piecewise recursive, since $B^{[e]} = \mathcal{W}_{f(e)}$.

Let $\mathcal{F}$ be the first-order formal system with equality, whose nonlogical symbols are the constants $\overline{n}$ and the unary predicate symbols P_n, for each n, and whose nonlogical axioms are:

- $\neg(\overline{x} = \overline{y})$, for $x \neq y$
- $P_e(\overline{x})$, for $x \in B^{[e]}$.

The set of axioms is r.e. because so is B, and thus $\mathcal{F}$ is a formal system.

It is obvious that every recursive set is weakly representable in $\mathcal{F}$: more precisely, $\mathcal{W}_{f(e)}$ is weakly representable by the formula $P_e(x)$. To prove the converse, i.e. that every weakly representable set is recursive, let $\varphi(x)$ be a unary formula containing only predicates among $P_0, \ldots, P_e$. We show how to decide whether any given instance $\varphi(\overline{x})$ is provable or not, by using only the columns $B^{[0]}, \ldots, B^{[e]}$: since these are recursive, so is the set weakly representable by φ.

The basic observation is the following:

- *a sentence of $\mathcal{F}$ is not provable if and only if it is false in some finite structure for $\mathcal{F}$ that does not make any axiom false*

> In one direction, if a sentence is not provable then it is false in some model of $\mathcal{F}$. Since there is no free variable, the sentence must then already be false in a finite substructure of the model.
>
> In the opposite direction, every finite structure that does not make any axiom false can be extended to a full model of $\mathcal{F}$. Thus any formula false in any such finite structure is false in a model, and it cannot be provable.

In particular, this proves that $\mathcal{F}$ is consistent. For example, $\neg(\overline{0} = \overline{0})$ is not provable, because it is false in any finite structure as above.

We first prove that the set of theorems of $\mathcal{F}$ is recursive in B. Since it is automatically r.e., because $\mathcal{F}$ is a formal system, it is enough to prove that its complement, i.e. the set of unprovable sentences, is r.e. in B. This follows from the observation above, since the set of finite structures that do not make any axiom false is r.e. in B.

Moreover, the set of theorems of $\mathcal{F}$ mentioning only predicates among $P_0, \ldots, P_e$ is recursive, because in this case it is enough to consider only the finite structures that interpret these predicates, and do not make any axiom false. The set of such structures is now r.e., since the last condition can be verified by checks only on the columns $B^{[0]}, \ldots, B^{[e]}$, which are recursive.

This concludes the proof of the first part. For the second part, we cannot claim directly that the set of theorems of $\mathcal{F}$ has incomplete degree, because the set B is creative (see III.2.6.a). However, we can build a similar formal system $\mathcal{F}_1$ by starting not from B, but from a thick r.e. subset A of B such that $\mathcal{K} \not\leq_T A$. A exists by the Thickness Lemma, since B is piecewise recursive.

To prove that all recursive sets are weakly representable in $\mathcal{F}_1$ it is not enough to consider only the formulas $P_e(x)$, since they represent only the columns $A^{[e]}$, which differ finitely from $B^{[e]}$. But to represent $B^{[e]}$ it is enough to add to $P_e(x)$ a finite disjunction of atomic formulas of the form $x = \overline{n}$, one for each $n \in B^{[e]} - A^{[e]}$.

The proof that all sets weakly representable in $\mathcal{F}_1$ are recursive, as well as the proof that the set of theorems of $\mathcal{F}_1$ is recursive in A and hence T-incomplete, are instead as above. $\Box$

Shoenfield [1961] proved that it is possible to strengthen the result by making all recursive functions strongly representable, and hence all recursive sets representable, in $\mathcal{F}$. This provides an example of a consistent formal system in which

$$\text{weakly representable} \Leftrightarrow \text{representable} \Leftrightarrow \text{recursive}.$$

For any consistent formal systems extending $\mathcal{R}$ one instead has, by II.2.16,

$$\begin{array}{rcl} \text{representable} & \Leftrightarrow & \text{recursive} \\ \text{weakly representable} & \Leftrightarrow & \text{recursively enumerable.} \end{array}$$

X.4 The Priority Method

The priority method is perhaps the most distinctive and basic tool of Recursion Theory. We thus stop for a moment to consider and discuss some methodological questions related to it, before plunging into its applications to the study of various degree structures.

Historical remarks ⋆

The priority method is, in its essence, a way to balance the individual actions needed to satisfy the various requirements of a construction, by means of a number of techniques: a global priority list of the requirements; a possible delay of the action needed to satisfy a given requirement, until appropriate conditions apply; and a possible undoing of the action already taken for the satisfaction of a given requirement, if and when the possibility of satisfying a requirement with higher priority arises.

A forerunner of some of these typical aspects of the priority method can be found in logic, in Ackermann's modification of Hilbert's ε-substitution method, that required the setting of an order of priority determined by the rank of an ε-term, whose value could change finitely often during the process (see Hilbert and Bernays [1939]).

In Recursion Theory the first appearance of the method was in Post's construction of a hypersimple set (Post [1944], see III.3.12), in the form of a *no-injury priority method*, in which at every stage one satisfies the requirement with smallest index that can be satisfied and has not yet been satisfied. This version of the priority method has been used repeatedly in Chapter VII, to prove results in Complexity Theory.

The *finite injury priority method* was introduced by Muchnik [1956] and Friedberg [1957a] for their solution to Post's Problem (X.1.7). Its essence has later been understood as an effective version of the finite extension (or the Baire Category) method, as explained in Section X.2.

The *infinite injury priority method* was introduced by Shoenfield [1961] and Sacks [1963b] to prove the Thickness Lemma (X.3.7) and the Sacks Jump Inversion Theorem (XI.1.23), respectively. The typical features of some of its applications have later been codified in the Injury and Window Lemmas, as explained in Section X.3.

The simple dichotomy between finite and infinite injury is however too narrow to capture the variety of priority arguments that have been used to prove advanced results in degree theory (most of which are outside the scope of this book). Among them one can quote the following:

- The *pinball machine method* (X.3.6), introduced by Lerman [1973] to extend the construction of minimal pairs from ω to some admissible ordinals, as well as to embed lattices in the r.e. degrees.

- The *monster injury priority method*, introduced by Lachlan [1975a] to prove the failure of Splitting and Density combined (X.6.16).
- The *tree method*, introduced by Lachlan [1975a] and Harrington [1982] and discussed in the next subsection, which allows a classification of the complexity of an argument in terms of the complexity of the computation of the final outcomes of the requirements in its construction.
- The *workers method*, introduced by Ash [1986] and Harrington to prove results in effective model theory. Its forerunner was the proof originally given by Lachlan [1965c] and Martin [1966b] of the existence of intermediate r.e. degrees (XI.1.22.c).

Various attempts have been made to produce a general framework for the priority method (Sacks [1963], Lachlan [1967a], [1970a], [1973], Yates [1974], Soare [1976], [1985], Lempp and Lerman [1990], [1995], [1997a], Knight [1990], Shoenfield [1990], Kontostathis [1991], [1992], [199?]). But, perhaps, the most illuminating point of view is that proposed by Harrington, according to which complicated priority arguments can (often) be seen as a sequence of simpler arguments on top of each other. For example, a finite injury on top of an infinite injury argument, as in Lachlan [1975a]. Or an infinite injury on top of a finite injury argument, as in Shore [1988a]. A good example related to this view is the proof by Jockusch and Shore [1983] of the Jump Hierarchy Theorem for r.e. degrees (see XI.1.19), in which the original infinite injury construction of an incomplete high r.e. degree is reduced to *two* successive finite injury arguments, and the original workers construction of an intermediate r.e. degree is reduced to *infinitely many* successive finite injury arguments, amalgamated by the Fixed-Point Theorem.

The Tree Method

A very useful framework for priority arguments, introduced by Lachlan [1975a] and developed by Harrington [1982], consists of dividing the description of an argument into two parts:

- a *local* description of the strategy needed to satisfy an individual requirement, dependent on a guess about the possible outcomes of the strategies for requirements of higher priority (for each requirement and any guess there must be a version of the strategy that would work);
- a *global* description of all possible outcomes of the individual strategies, arranged on a tree.

In the simplest cases the only relevant outcome of strategies for requirements of higher priority is simply whether they act or not. By coding these two possibilities as 0 and 1, the tree of all possible outcomes thus becomes the usual binary tree. In more complicated cases one may need to know more about requirements of higher priority than simply whether they act or not, e.g. the actual values of the restraint function imposed by them. By coding this information as numbers, the tree of all possible outcomes then becomes a tree in the sense of IV.2.14, i.e. a set of sequence numbers. In still more complicated cases one may need to know even more information about requirements of higher priority, which may require ordinals less than α to code, i.e. α-branching trees, for some ordinal α.

The first historical examples of arguments that, at least implicitly, used trees as above have been:

- for $\alpha = 2$, the maximal set construction of Friedberg [1958] (III.4.18)
- for $\alpha = \omega$, the minimal pair construction of Lachlan [1966b] and Yates [1966] (X.6.5)
- for $\alpha = \omega + 1$, a jump noninversion theorem of Shore [1988a].

We now provide two examples of tree arguments, respectively with $\alpha = 2$ and $\alpha = \omega$, by adapting the proofs of the two central results of Sections 1 and 3. The first example is a tree version of a finite injury priority argument.

Proposition X.4.1 A solution to Post's Problem by the Tree Method. *An r.e. set which is neither recursive nor T-complete can be constructed by the tree method.*

Proof. We adapt the proof of X.1.2 and keep the same notation as there. We build two r.e. sets A and B such that A is coinfinite and the following requirements are satisfied:

$$\begin{aligned} P_e &\ : \ \mathcal{W}_e \text{ infinite} \Rightarrow \mathcal{W}_e \cap A \neq \emptyset \\ N_e &\ : \ B \not\simeq \{e\}^A. \end{aligned}$$

We say that the requirement P_e *is active* at stage s if $\mathcal{W}_{e,s} \cap A_s \neq \emptyset$. To *become active* at stage s the requirement has to take into account the restraints imposed by negative requirements of higher priority, i.e. it can only put into A an element bigger than $\max_{i \le e} r(i, s-1)$. However, once the requirement P_e has become active it remains so forever, since if $\mathcal{W}_{e,s} \cap A_s \neq \emptyset$, then $\mathcal{W}_e \cap A \neq \emptyset$. Thus P_e may only switch from being inactive to being active, but not conversely.

Similarly, we say that the negative requirement N_e *is active* at stage s if $\{e\}_s^{A_s}(z_e^s) \simeq 0$ and $z_e^s \in B_s$, in which case it imposes a restraint $r(e, s)$

to preserve the computation (by convention, $r(e, s) = 0$ if N_e is inactive). To *become active* at stage s the requirement does not have to take into account the positive requirements of higher priority. However, since positive requirements of higher priority can later injure it and force a change in z_e^s, the requirement N_e may not only switch from being inactive to being active, but also conversely (although only finitely often).

We can code partial information about whether positive or negative requirements are active by strings σ of 0's and 1's, with the following conventions: $\sigma(2e)$ refers to the negative requirement N_e; $\sigma(2e+1)$ refers to the positive requirement P_e; the value of $\sigma(n)$ is 0 or 1, according to whether the corresponding requirement is active or not. We say that σ *looks correct* at stage s if its values reflect the current situation at stage s, i.e. they are 0 if the corresponding requirement is active at stage s and 1 otherwise; and we say that σ *is correct* if its values reflect the final situation, i.e. they are 0 if there is a stage s after which the corresponding requirement is permanently active and 1 otherwise. The correct strings define the *true path* of the construction, while the strings that look correct at some stage define partial approximations to the true path.

We can visualize the true path among its approximations in terms of the following ordering, which orders compatible strings by extension, and incompatible strings as in the Kleene-Brouwer ordering (p. I.386), i.e. the order induced by the condition $0 < 1$ on the first place in which the strings disagree:

$$\sigma \prec \tau \;\Leftrightarrow\; \sigma \subset \tau \;\vee\; (\exists n)[\sigma(n) < \tau(n) \;\wedge\; (\forall i < n)(\sigma(i) = \tau(i))].$$

Since what the construction of X.1.2 really does is to try to activate as many requirements as possible, i.e. to switch numbers from 1 to 0 in the string coding the current situation, what happens is that any time a requirement becomes active the current approximation to the true path moves towards the left. Moreover, positive requirements never switch back from active to inactive; and negative requirements do so only when some positive requirement of higher priority injures them, in which case again the current approximation moves to the left (because it changes from 1 to 0 for a requirement with smaller index). In other words, the strings defining the true path are the leftmost ones in the partial subtree of its approximations.

The present construction is similar to that of X.1.2, with the difference that we now use strings σ as indices of requirements, as follows:

$$R_\sigma = \begin{cases} N_e & \text{if } |\sigma| = 2e \\ P_e & \text{if } |\sigma| = 2e+1. \end{cases}$$

Instead of having requirements indexed by numbers e, that can be satisfied at most 2^e times because of injuries, we thus now have requirements indexed by strings σ, that can be satisfied only once. However, nothing essential has changed, since the requirement R_σ really depends only on $|\sigma|$ and thus appears in the list $2^{|\sigma|}$ times. Moreover, instead of saying that one requirement indexed by a given number is satisfied finitely many times, we say that finitely many requirements indexed by given strings having the same length are each satisfied at most once.

The order of priority among requirements is naturally translated as follows: R_σ has higher priority than R_τ if $\sigma \prec \tau$, i.e. either σ is extended by τ, or it lies to the left of it. The first condition states that requirements with smaller indices have higher priority, while the second condition states that one tries to satisfy a requirement (and hence to move left w.r.t. $\prec$) whenever possible.

We are now ready to rephrase the construction of X.1.2 in the present framework. We begin by letting

$$A_0 = B_0 = \emptyset \quad z_\sigma^0 = \langle \sigma, 0 \rangle \qquad \text{and} \qquad r(\sigma, 0) = 0,$$

where σ is seen as a sequence number in $\langle \sigma, 0 \rangle$.

At stage $s+1$, given A_s, B_s, z_σ^s and $r(\sigma, s)$, consider the strings σ such that

- $|\sigma| \leq s$
- σ looks correct at stage s
- R_σ is inactive at stage s but can become active, i.e.
    1. if $|\sigma| = 2e$, $\{e\}_s^{A_s}(z_\sigma^s) \simeq 0$ and $z_\sigma^s \notin B_s$
    2. if $|\sigma| = 2e+1$, $\mathcal{W}_{e,s} \cap A_s = \emptyset$ and there is an x such that

$$x \in \mathcal{W}_{e,s} \ \wedge \ x \geq 2e \ \wedge \ (\forall \tau \prec \sigma)(x > r(\tau, s)).$$

If such a string σ exists, take the smallest one (i.e. consider the requirement R_σ of highest priority that is inactive but can become active). Moreover,

1. in case 1 put z_σ^s into B and let $r(\sigma, s+1) = u(e, z_\sigma^s, s)$
2. in case 2 put the smallest such x into A, and for every string τ such that $\sigma \prec \tau$ let $r(\tau, s+1) = 0$ and

$$z_\tau^{s+1} = \mu y. \, (\exists z \leq y)(\langle \tau, z \rangle = y \ \wedge \ y > z_\tau^s).$$

All other witnesses and restraints are left unchanged.

It is clear that $A = \bigcup_{s\in\omega} A_s$ and $B = \bigcup_{s \in \omega} B_s$ are r.e. and $\overline{A}$ is infinite (by the condition $x \geq 2e$). To show that all requirements N_e and P_e are satisfied, it is enough to prove that:

- *R_σ is satisfied, for each σ on the true path*
  Given σ on the true path, there is a stage s_0 such that σ looks correct at every stage $s \geq s_0$. Indeed, by construction the approximations to the true path only move left w.r.t. $\prec$, so that once the true path is reached the approximations become final. This means that

  $$z_\sigma = \lim_{s\to\infty} z_\sigma^s = z_\sigma^{s_0},$$

  and for any $\tau \subset \sigma$

  $$r(\tau) = \lim_{s\to\infty} r(\tau, s) = r(\tau, s_0).$$

  The proof that R_σ is satisfied can now continue exactly as in X.1.2, according to whether $|\sigma| = 2e$ (i.e. $R_\sigma = N_e$) or $|\sigma| = 2e+1$ (i.e. $R_\sigma = P_e$). □

In the previous construction *the true path t is recursive in $\emptyset'$*. To prove this, it is enough to show how to compute $t(n)$ recursively in $\emptyset'$, by induction on n. Given σ such that

$$\sigma(i) = \begin{cases} t(i) & \text{if } i < n \\ \text{undefined} & \text{otherwise,} \end{cases}$$

let s_0 be a stage such that σ looks correct at stage s_0. Then

$$t(n) = \begin{cases} 0 & \text{if } (\exists s \geq s_0)(R_\sigma \text{ is active at stage } s) \\ 1 & \text{otherwise.} \end{cases}$$

Since both the construction and the check of whether a given requirement is active at a given stage are recursive, to compute $t(n)$ we only need an oracle for one quantifier. Thus t is recursive in $\emptyset'$.

Exercise X.4.2 *A maximal set can be constructed by the tree method.* (Hint: we adapt the proof of III.4.18, keeping the same notation. We build a coinfinite r.e. set A satisfying the following requirements:

$$\begin{aligned} P_e \quad : \quad & \mathcal{W}_e \cap \overline{A} \text{ finite or } \overline{\mathcal{W}}_e \cap \overline{A} \text{ finite,} \\ & \text{i.e. } \mathcal{W}_e \cap \overline{A} \text{ infinite } \Rightarrow \overline{A} \subseteq^* \mathcal{W}_e. \end{aligned}$$

Strings of 0's and 1's code e-states, with the convention that if $i \leq e < |\sigma|$, then $\sigma(i)$ is 0 or 1 according to whether $a_e \in \mathcal{W}_i$ or $a_e \notin \mathcal{W}_i$. We let $R_\sigma = P_{|\sigma|-1}$. The construction at stage $s+1$ simply tries to move the e-state of a_e^{s+1} to the left w.r.t. $\prec$, whenever possible. The proof of III.4.8 then shows that R_σ is satisfied, for each σ on the true path.)

We now turn to our second example, which is a tree version of an infinite injury priority argument.

Proposition X.4.3 The Thickness Lemma by the Tree Method. *Given a nonrecursive r.e. set C and an r.e. set B such that $C \not\leq_T B^{[<e]}$ for every e, a thick subset A of B such that $C \not\leq_T A$ can be constructed by the tree method.*

Proof. We adapt the proof of X.3.7 and keep the same notation as there. We build an r.e. set A such that the following requirements are satisfied:

$$\begin{aligned} P_e &: B^{[e]} \subseteq^* A \\ N_e &: C \not\simeq \{e\}^A. \end{aligned}$$

We say that the requirement P_e *is active* at stage s if $A_s^{[e]} - A_{s-1}^{[e]} \neq \emptyset$, i.e. if some element of $B_s^{[e]}$ goes into A at stage s. To *become active* at stage s the requirement has to take into account the restraints imposed by negative requirements of higher priority, i.e. it can only put into A the elements of $B^{[e]}$ that are bigger than $\max_{i \leq e} \hat{r}(i, s-1)$. Moreover, since $B^{[e]}$ can be infinite and we have no control on how it is generated, P_e may switch back and forth between being active and inactive. The relevant outcome of P_e is thus whether it is active infinitely often or not.

The negative requirement N_e is instead always active at stage s, since its role is to impose a restraint $\hat{r}(e, s)$ to preserve a number of agreements, as well as the first disagreement, between current values of C and $\{e\}^A$. However, since positive requirements of higher priority can later injure it and they are infinitary, the restraint imposed by N_e can change infinitely often. The relevant outcome of N_e is thus the value of $\liminf_{s\to\infty} \hat{r}(e, s)$.

We can code partial information about the outcomes of positive or negative requirements by strings σ, with the following conventions: $\sigma(2e)$ refers to the negative requirement N_e and it is an integer guessing the value of $\liminf_{s\to\infty} \hat{r}(e, s)$; $\sigma(2e+1)$ refers to the positive requirement P_e, and it is 0 or 1 according to whether we guess that the requirement will become active infinitely or finitely often (the tree of possible outcomes is thus alternately ω-branching and binary). We say that σ *looks correct* at stage s if its values reflect the current situation at stage s; and we say that σ *is correct* if its values reflect the final situation, i.e. the actual outcomes of the requirements. The correct strings define the *true path* of the construction, and the strings that look correct at some stage define partial approximations to the true path.

In terms of the ordering $\prec$ among strings introduced in X.4.1, any time a positive requirement that was not active becomes so, or a negative requirement imposes a smaller restraint than it did before, the current approximation to the true path moves towards the left. However, since the positive requirements are infinitary, the approximations may switch back and forth. Thus, by the definition of the relevant outcomes, the true path is no longer the leftmost one in the partial subtree of its approximations, but only the leftmost one that looks correct infinitely often.

The present construction is similar to that of X.3.7, with changes similar to those introduced in X.4.1. In other words, we use strings as indices of requirements, as follows:

$$R_\sigma = \begin{cases} N_e & \text{if } |\sigma| = 2e \\ P_e & \text{if } |\sigma| = 2e+1, \end{cases}$$

and the order of priority among requirements is as follows: R_σ has higher priority than R_τ if $\sigma \prec \tau$, i.e. either σ is extended by τ, or it lies to the left of it.

We are now ready to rephrase the construction of X.3.7 in the present framework. We begin by letting $A_0 = B_0 = \emptyset$. At stage $s+1$, given A_s and C_s, for any e let

$$\begin{array}{rcl} \hat{l}(e,s) & = & \max\{z : (\forall y < z)[C_s(y) \simeq \{\hat{e}\}_s^{A_s}(y)]\} \\ \hat{r}(e,s) & = & \max\{\hat{u}(e,x,s) : x \leq \hat{l}(e,s)\}, \end{array}$$

where $\{\hat{e}\}_s^{A_s}$ is defined as in X.3.4. Consider the unique string σ_s of length s that looks correct at stage s, i.e. such that on its domain:

- $\sigma_s(2e) = \hat{r}(e,s)$, i.e. σ_s is guessing the current restraint for the negative requirement N_e;
- $\sigma_s(2e+1) = 0$ if some element of the e-th column has been enumerated in A after the last stage $t < s$ at which σ_t agreed with σ_s below $2e+1$ (if there is one), and $\sigma_s(2e+1) = 1$ otherwise.

Then, for any e such that $2e+1 \leq s$ we let

$$x \in A_{s+1} \Leftrightarrow x \in B_{s+1}^{[e]} \wedge (\forall i \leq e)(\forall t \leq s)(\sigma_t \preceq \sigma_s \Rightarrow x > \sigma_t(2i)),$$

i.e. we take into account not only the current guess, but also the previous ones to the left of it (since they might be on the true path).

To show that all requirements N_e and P_e are satisfied, it is enough to prove that:

- *R_σ is satisfied, for each σ on the true path*
  Given σ on the true path, there are infinitely many stages s such that σ looks correct at stage s. Indeed, by construction the approximations to the true path move left w.r.t. $\prec$ every time some positive requirement becomes active, or the restraint of a negative requirement drops down.

  The proof that R_σ is satisfied can now continue exactly as in X.3.7, according to whether $|\sigma| = 2e$ (i.e. $R_\sigma = N_e$, in which case the Injury

Lemma is used), or $|\sigma| = 2e+1$ (i.e. $R_\sigma = P_e$, in which case the Window Lemma using the true stages of the construction is not needed because σ is on the true path, and hence the restraints of all negative requirements of higher priority simultaneously drop back to their lim inf). $\square$

In the previous proof the tree method replaces the true stages trick and the Window Lemma, while the hat trick and the Injury Lemma are still needed to ensure $C \not\leq_T A$ when C is a given nonrecursive r.e. set. However, in constructions in which C is built simultaneously with A, as opposed to being given ahead of time, or when A is simply required to be nonrecursive, the tree method avoids the hat trick and the Injury Lemma too, thus becoming a complete and more flexible alternative to the methods of Section 3.

In the previous construction *the true path t is recursive in $\emptyset''$*. This is immediate by definition, i.e.

$$t(n) = \mu x.\,(\exists_\infty s)(\sigma_s(n) = x).$$

Since the construction and the definition of σ_s are recursive and $\exists_\infty s$ is equivalent to $\forall y \exists s > y$, to compute $t(n)$ we only need an oracle for two quantifiers and thus t is recursive in $\emptyset''$.

In most of the advanced applications of the tree method the true path is still defined as above and it is recursive in $\emptyset''$ for the same reason.[1] But it may no longer be true that R_σ is satisfied for each σ on the true path. Unlike in the two examples above, in which the computations of the final outcomes and of the true path had the same complexity, in general it could be more complicated to compute the final outcomes of the requirements than the true path (e.g. because there could be more than one strategy for each requirement on the true path, some of which might be injured; these arguments thus take the form of *a finite injury argument along the true path*, or *a finite injury argument on top of an infinite injury one*). The complexity of the computation of the final outcomes of the requirements is thus a possible measure of the complexity of the corresponding tree argument.

Tree arguments whose such complexity is recursive in $\emptyset'$, $\emptyset''$ and $\emptyset'''$ correspond to the finite, infinite and monster injury priority arguments discussed in the historical remarks. But the measure of the complexity of the computation of the final outcomes allows for a finer classification of priority arguments. For example, Shore [1988a] has proved that, for every degree $\boldsymbol{a}$ r.e. in and above $\mathbf{0}''$, there is a tree argument whose complexity has exactly degree $\boldsymbol{a}$.

For more detailed expositions of the tree method see Soare [1985], [1987] and Downey [1990].

[1]However, most does not mean all. For example, in Shore [1988a] the true path is only recursive in $\emptyset'''$.

Admissibility ⋆

One possible way, and historically the first one, of classifying axiomatically the strength of the priority arguments needed to prove results about r.e. degrees consists in seeing ω as an ordinal and classifying the amount of closure properties needed to be able to reproduce the arguments on the one hand, and to prove the results (possibly by different arguments) on the other.

Following the definitions on p. I.443, we say that: an ordinal α is Σ_n-*admissible* if it is closed under the $\Sigma_n^{L_\alpha}$-definable functions (Σ_1-admissible ordinals were simply called admissible on p. I.443); a function on α is α-*recursive* if it is $\Sigma_1^{L_\alpha}$-definable; a subset of α is α-*recursive* if it is $\Delta_1^{L_\alpha}$-definable or, equivalently, if it has an α-recursive characteristic function; and a subset of α is α-*recursively enumerable* (α-r.e.) if it is $\Sigma_1^{L_\alpha}$-definable or, equivalently, if it is the range of an α-recursive function.

A basic role is played by the following notion, patterned on IV.3.22.1: a subset of α is α-*finite* if it belongs to L_α or, equivalently, if it is α-recursive and α-bounded (i.e. contained in an ordinal strictly less than α). As in the proof of IV.3.22, if α is Σ_1-admissible, then there is an α-recursive enumeration $\{D_x\}_{x\in\alpha}$ of the α-finite subsets of α.

The appropriate notion of relative recursiveness is patterned on III.1.4. Given two subsets A and B of α, A is α-*recursive in* B ($A \leq_\alpha B$) if, for some α-r.e. relation R,

$$D_y \subseteq A \ \wedge\ D_x \subseteq \overline{A} \ \Leftrightarrow\ (\exists u, v)(D_v \subseteq B \ \wedge\ D_u \subseteq \overline{B} \ \wedge\ R(x, y, u, v)).$$

For A and B α-r.e. this reduces, as in III.1.4, to

$$D_x \subseteq \overline{A} \ \Leftrightarrow\ (\exists u)(D_u \subseteq \overline{B} \ \wedge\ R(x, u)).$$

$\leq_\alpha$ is a reflexive and transitive relation (transitivity being ensured by the fact that α-finite sets are used both on the left and on the right),[2] and it thus induces as usual an equivalence relation

$$A \equiv_\alpha B \ \Leftrightarrow\ A \leq_\alpha B \ \wedge\ B \leq_\alpha A,$$

whose equivalence classes are called (α-r.e.) α-*degrees*.

[2]Driscoll [1968] has proved that the notion of relative α-recursiveness obtained by using only single elements on the left is not in general transitive for Σ_1-admissible ordinals. The difficulty is that the first reduction associates to every element x an α-finite set D and the second reduction associates to every element of D an α-finite set, but the association is not necessarily α-recursive, so that the set of α-finite sets corresponding to elements of D is not necessarily α-finite, and one cannot put together all the α-finite sets corresponding to elements of D into a single α-finite set corresponding to x. Shore [1975a] has characterized the Σ_1-admissible ordinals for which the problem does not arise.

By IV.3.16, if α is Σ_1-admissible, then there is a $\Sigma_1^{L_\alpha}$-enumeration $\{\mathcal{W}_x\}_{x\in\alpha}$ of the α-r.e. sets, and hence an α-complete α-r.e. set which is not α-recursive by IV.3.18. Thus there are at least two α-r.e. α-degrees $\mathbf{0}_\alpha$ and $\mathbf{0}'_\alpha$. Post's Problem asks whether there are others.

Theorem X.4.4 Solution to Post's Problem for Σ_1-admissible ordinals (Sacks [1966a]) *For any Σ_1-admissible ordinal α, there exists an α-r.e. set which is neither α-recursive nor α-complete.*

Proof. We adapt the proof of X.1.2 and keep the same notation as there, with the understanding that the variables no longer range over finite numbers, i.e. ordinals less than ω, but range over ordinals less than α. In particular, both the indices e of the α-recursive reductions and the stages s of the construction will be ordinals less than α.

We build two α-r.e. sets A and B such that $\overline{A}$ is α-unbounded and the following requirements are satisfied:

$$\begin{aligned} P_e &: \mathcal{W}_e \ \alpha\text{-unbounded} \Rightarrow \mathcal{W}_e \cap A \neq \emptyset \\ N_e &: B \not\simeq \{e\}^A. \end{aligned}$$

The main problem is to ensure that the negative requirements are injured only α-finitely often. The solution is as follows. Notice that a positive requirement can injure N_e at most once, because when it is satisfied it remains so forever. It is thus enough to show that the set of indices of positive requirements injuring N_e is α-finite. This set is α-r.e. because it can be recovered from the construction, and it is α-bounded because a positive requirement can injure N_e only if it has an index less than e. Unfortunately, being α-r.e. and α-bounded is somewhat less than being α-finite (i.e. α-recursive and α-bounded). But since the sets we are considering are sets of indices, one solution to the problem is to make the list of indices so short that every α-r.e. α-bounded subset of it is α-finite.

We define the *Σ_1-projectum α^* of* α as the smallest ordinal $\beta \leq \alpha$ such that there is a one-one α-recursive function $f : \alpha \to \beta$. The following is the crucial property we are looking for:

- *an α-r.e. α^*-bounded set is α-finite*

  It is enough to prove the analogue of II.1.18.a, i.e. that any α-infinite α-r.e. set C is the range of a one-one α-recursive function f. If β is the l.u.b. of C, then $\alpha^* \leq \beta$ by definition of α^* and hence C cannot be α^*-bounded.

  To define f, let g be an α-recursive function with range C. Start with $f(0) = g(0)$. For any stage $s > 0$, let $f(s)$ be the first element generated by g that is not yet in the range of f. If f were not total, then there would be a stage $s < \alpha$ such that C is the image of g restricted to s. Thus C would be α-finite (because, by definition of Σ_1-admissibility, L_α is closed under α-recursive functions and if $s < \alpha$, then $s \in L_\alpha$).

The basic use of the Σ_1-projectum is to shorten the lists $\{\mathcal{W}_e\}_{e\in\alpha}$ of the α-r.e. sets and $\{\{e\}^X\}_{e\in\alpha}$ of the functions α-recursive in X (and hence of the positive and negative requirements) to lists $\{\mathcal{W}_e\}_{e\in\alpha^*}$ and $\{\{e\}^X\}_{e\in\alpha^*}$, via any one-one α-recursive function $f : \alpha \to \alpha^*$ (whose existence is ensured by the definition of α^*).

The shortening of the index list from α to α^* requires a small adjustment in the construction of a set with α-unbounded complement. It is not enough to let an element x go into A for the sake of the positive requirement P_e only if $x > 2e$, since now $e < \alpha^*$, i.e. $2e < 2\alpha^*$, and $2\alpha^*$ could very well be much smaller than α (in which case, any element greater than $2\alpha^*$ would be permitted to go into A). Instead of letting x enter A only if it is greater than twice the *indices* of requirements of higher priority, we will thus let x enter A only if it is greater than twice the *elements* that have already entered A for the sake of requirements of higher priority.

The construction is similar to that of X.1.2 (in particular we use an α-recursive pairing function to split α into columns), with the small difference that now we have not only successor stages, but also limit ones. We begin by letting

$$A_0 = B_0 = \emptyset \qquad z_e^0 = \langle e, 0\rangle \qquad \text{and} \qquad r(e,0) = 0.$$

At limit stages s we let

$$A_s = \bigcup_{t<s} A_t, \qquad B_s = \bigcup_{t<s} B_t \qquad z_e^s = \lim_{t\to s} z_e^t \qquad \text{and} \qquad r(e,s) = \lim_{t\to s} r(e,t).$$

Notice that these limits exist because the functions z_e^t and $r(e,t)$ are α-recursive by construction, s is less than α and α is Σ_1-admissible, i.e. closed under the α-recursive functions.

At successor stages $s+1$, given A_s, B_s, z_e^s and $r(e,s)$, the construction is as follows:

- *B-part*
  For each $e \leq s$, if $\{e\}_s^{A_s}(z_e^s) \simeq 0$ and $z_e^s \notin B_s$, put z_e^s into B.
- *A-part*
  For each $e \leq s$, if $\mathcal{W}_{e,s} \cap A_s = \emptyset$ and for some x:
  - $x \in \mathcal{W}_{e,s}$
  - $x \geq 2y$, for any y that has ever entered A_s for the sake of some positive requirement with index $< e$

– $(\forall i \leq e)(x > r(i,s))$,

then take the smallest such x and put it into A.

Moreover, if N_e has been injured at step $s+1$, then we change z_e^s to the next bigger element in the e-th column of α. Finally, we let

$$A = \bigcup_{s \in \alpha} A_s \qquad \text{and} \qquad B = \bigcup_{s \in \alpha} B_s.$$

It is clear that A and B are α-r.e. by definition.

- *$\overline{A}$ is α-unbounded*
  Suppose $\overline{A}$ is α-bounded. Then there is an $a < \alpha$ such that all $x \geq a$ are in A. We construct an infinite descending sequence $\{e_n\}_{n \in \omega}$ in α, contradicting the fact that α is an ordinal and hence it is well-founded.

  Let a_0 be any infinite element greater than a and than every element of A_s, where s is the stage such that a_0 enters A at stage $s+1$ (i.e. $a_0 \in A_{s+1} - A_s$). Given a_n, let a_{n+1} be the first element x generated in A after a_n and such that $a_0 < x < 2a_n$. Such an x exists, by the choices of a and a_0.

  Letting e_n be the index of the positive requirement for whose satisfaction a_n goes into A, we want to show that $e_{n+1} < e_n$. Since a_n has entered A for the satisfaction of the positive requirement with index e_n, the construction allows only elements $\geq 2a_n$ to later enter A for the sake of positive requirements with index $> e_n$. Thus a_{n+1} (which is $< 2a_n$) must have entered for the sake of a positive requirement with index $e_{n+1} < e_n$ (e_{n+1} cannot be equal to e_n, because the positive requirements are satisfied at most once).

- *N_e is satisfied, for each $e \in \alpha^*$*
  Notice that N_e can be injured only α-finitely many times. Indeed, any positive requirement can injure it at most once and the set of indices of positive requirements that injure it is α-r.e. (by construction) and α^*-bounded (below e), hence α-finite.

  The proof of the satisfaction of N_e can thus proceed as in X.1.2.

- *$\lim_{s \to \alpha} r(e,s) < \alpha$, for each $e \in \alpha^*$*
  As in X.1.2.

- *P_e is satisfied, for each $e \in \alpha^*$*
  Let

$$\begin{aligned} r(e) &= \lim_{s \to \alpha} r(e,s) \\ R(e) &= \max_{i \leq e} r(i). \end{aligned}$$

Notice that $R(e)$ is defined because α is Σ_1-admissible and hence it is closed under α-recursive functions. Let s_0 be a stage such that

$$(\forall s \geq s_0)(\forall i \leq e)[r(i,s) = r(i,s_0)],$$

and such that all elements y that ever enter A for the sake of some positive requirement with index $< e$ have already entered by stage s_0.

If $\mathcal{W}_e$ is α-unbounded, there is an x such that

$$x \in \mathcal{W}_e \ \wedge \ x \geq \max\{2y : y \in A_{s_0}\} \ \wedge \ x > R(e).$$

So either $\mathcal{W}_{s_0} \cap A_{s_0} \neq \emptyset$ or, as soon as one of those x's appears in $\mathcal{W}_e$ after stage s_0, it goes into A. $\square$

Sacks and Simpson [1972] and Shore [1975] have extended the previous proof to obtain (in a quite more complicated way) the analogues of all results in Section X.1. One could thus guess that Σ_1-admissibility on the ordinals is an appropriate abstract setting for finite injury priority arguments. Instead, it turns out that it is stronger than needed. On the one hand, one can also solve Post's Problem on some (although not all) admissible structures that are not well-ordered (Simpson [1974], Harrington), as well as on some (although not all) ordinals that are not Σ_1-admissible (Friedman and Sacks [1977]). On the other hand, on Σ_1-admissible ordinals one can also carry out some typical infinite injury argument, such as a proof of the Density Theorem (Shore [1976]).

However, Σ_1-admissibility is not sufficient to carry out all infinite injury arguments. For example, *there are Σ_1-admissible ordinals α such that every incomplete α-r.e. α-degree is low*, e.g. $\alpha = \omega_\omega^L$, although for every Σ_2-admissible ordinal α there are incomplete high α-r.e. α-degrees (Shore [1976a]). A complete characterization of the ordinals α for which incomplete high α-r.e. α-degrees exist has been obtained, in terms of the relationship between various cofinalities and projecta (Maass [1978]).

One can also define natural analogues of the lattice-theoretical properties of r.e. sets. First, as in IX.3.1, a subset of α is said to be *α^*-finite* if every α-r.e. subset of it is α-recursive (the name obviously comes from the fact that the notion is equivalent to α^*-finiteness for the admissible ordinal α^* used in the proof of X.4.4). Then one says that a co-α^*-infinite α-r.e. set A is: *α-simple* if every α-r.e. subset of its complement is α^*-finite; *α-hyperhypersimple* if the lattice of its α-r.e. supersets is a Boolean algebra; and *α-maximal* if it has only trivial α-r.e. supersets (i.e. every such superset differs α^*-finitely from either A or α).

While α-simple sets exist for every Σ_1-admissible ordinal (by the proof of X.4.4), all the following possibilities are realized: there is no α-hyperhypersimple set, e.g. for $\alpha = \omega_1^L$; there are α-hyperhypersimple sets, but no maximal set, e.g. for $\alpha = \omega_\omega^L$; and there are α-maximal sets, e.g. for $\alpha = \omega$ (Lerman and Simpson [1973], Chong and Lerman [1976]). A complete characterization of the Σ_1-admissible ordinals α for which α-maximal sets exist has been obtained, as the class of countable ordinals α for which there exists an onto function $f : \omega \to \alpha$ that is the double limit of an α-recursive function (Lerman [1974a]).

The results just quoted show that, even by restricting attention to ordinals, admissibility assumptions provide only a rough classification of the strength needed to carry out priority arguments and to prove degree-theoretical results. In particular, additional notions such as cofinalities and various kinds of projecta are needed to obtain characterizations of the structural properties needed in specific cases.

For surveys of the present topic see Shore [1977a] and Chong and Friedman [1999]. For detailed treatments, see Chong [1984] and Sacks [1990].

Bounded Arithmetic ⋆

A different but related way of classifying axiomatically the strength of the priority arguments needed to prove results about r.e. degrees consists of seeing ω not as an ordinal but as the standard model of Peano Arithmetic $\mathcal{PA}$, thus classifying the amount of induction needed to be able to reproduce the arguments on the one hand, and to prove the results (possibly by different arguments) on the other.

Given a structure $\mathcal{M}$ for arithmetic, we say that $\mathcal{M}$ is a model of Σ_n-*induction* if it satisfies the Axiom of Induction (p. I.20)

$$\varphi(0) \wedge (\forall x)[\varphi(x) \to \varphi(\mathcal{S}(x))] \ \to \ (\forall y)\varphi(y)$$

restricted to Σ_n^0 formulas φ. Similarly, we say that $\mathcal{M}$ is a model of Σ_n-*collection* if it satisfies the Axiom of Collection (p. I.398)

$$(\forall y)[(\forall x < y)(\exists z)\varphi(x,z) \ \to \ (\exists a)(\forall x < y)(\exists z < a)\varphi(x,z)]$$

restricted to Σ_n^0 formulas φ. A natural consequence of Σ_n-collection is closure of the Σ_n^0-formulas under bounded (universal) quantification.

While the two Axioms of Induction and Collection are globally equivalent, locally they are not. The relationship among their restricted versions is summarized as follows:

$$\Sigma_{n+1}\text{-induction} \ \Rightarrow \ \Sigma_{n+1}\text{-collection} \ \Rightarrow \ \Sigma_n\text{-induction}$$

(see the proof of X.4.5 for the first implication).

A function on $\mathcal{M}$ is $\mathcal{M}$-*recursive* if it is $\Sigma_1^{\mathcal{M}}$-definable. A subset of $\mathcal{M}$ is: $\mathcal{M}$-*recursive* if it is $\Delta_1^{\mathcal{M}}$-definable; $\mathcal{M}$-*recursively enumerable* ($\mathcal{M}$-r.e.) if it is

$\Sigma_1^{\mathcal{M}}$-definable; and $\mathcal{M}$-*finite* if it can be coded by an element of $\mathcal{M}$, under some standard $\mathcal{M}$-recursive coding function.

The simplest setting in which Basic Recursion Theory can be developed is on models $\mathcal{M}$ of some suitable subsystem $\mathcal{PA}^-$ of Primitive Recursive Arithmetic, strong enough to prove that there are $\mathcal{M}$-recursive enumerations $\{D_x\}_{x\in\mathcal{M}}$ of the $\mathcal{M}$-finite sets and $\{\mathcal{W}_x\}_{x\in\mathcal{M}}$ of the of the $\mathcal{M}$-r.e. sets, and hence an $\mathcal{M}$-complete $\mathcal{M}$-r.e. set which is not $\mathcal{M}$-recursive (basically, one only needs axioms for the usual properties of sum, product and exponentiation, as well as induction for quantifier-free formulas).

Given two subsets A and B of $\mathcal{M}$, A is $\mathcal{M}$-*recursive in* B ($A \leq_{\mathcal{M}} B$) if, for some $\mathcal{M}$-r.e. relation R,

$$D_y \subseteq A \ \wedge\ D_x \subseteq \overline{A} \ \Leftrightarrow\ (\exists u, v)(D_v \subseteq B \ \wedge\ D_u \subseteq \overline{B} \ \wedge\ R(x, y, u, v)).$$

For A and B $\mathcal{M}$-r.e. this reduces as usual to

$$D_x \subseteq \overline{A} \ \Leftrightarrow\ (\exists u)(D_u \subseteq \overline{B} \ \wedge\ R(x, u)).$$

$\leq_{\mathcal{M}}$ is a reflexive and transitive relation (transitivity being ensured by the fact that $\mathcal{M}$-finite sets are used both on the left and on the right),[3] and it induces as usual an equivalence relation

$$A \equiv_{\mathcal{M}} B \ \Leftrightarrow\ A \leq_{\mathcal{M}} B \ \wedge\ B \leq_{\mathcal{M}} A,$$

whose equivalence classes are called ($\mathcal{M}$-r.e.) $\mathcal{M}$-*degrees*. On models of $\mathcal{PA}^-$ there are thus at least two $\mathcal{M}$-r.e. $\mathcal{M}$-degrees $\mathbf{0}_{\mathcal{M}}$ and $\mathbf{0}'_{\mathcal{M}}$. Post's Problem asks whether there are others.

Theorem X.4.5 Solution to Post's Problem for models of $\boldsymbol{\Sigma_1}$-induction (Simpson) *For any model $\mathcal{M}$ of Σ_1-induction, there exists an $\mathcal{M}$-r.e. set which is neither $\mathcal{M}$-recursive nor $\mathcal{M}$-complete.*

Proof. We only need to prove that a model $\mathcal{M}$ of Σ_1-induction shares the properties of Σ_1-admissible ordinals that make the proof of X.4.4 go through, namely:

- *$\mathcal{M}$ is closed under the $\mathcal{M}$-recursive functions*
  Since closure under the $\mathcal{M}$-recursive functions is a version of $\Sigma_1^{\mathcal{M}}$-replacement (see p. I.398) and collection is simply a form of replacement that avoids the mention of functions, we actually prove that Σ_1-induction implies Σ_1-collection.

[3]Groszek and Slaman [199?] have proved that the notion of relative $\mathcal{M}$-recursiveness obtained by using only single elements on the left is not in general transitive for models of Σ_1-induction.

Given a Σ_1^0 formula φ, we prove

$$(\forall y)[(\forall x < y)(\exists z)\varphi(x,z) \ \rightarrow \ (\exists a)(\forall x < y)(\exists z < a)\varphi(x,z)].$$

We suppose

$$(\forall x < y)(\exists z)\varphi(x,z),$$

and prove

$$(\forall t)[t \leq y \ \rightarrow \ (\exists a)(\forall x < t)(\exists z < a)\varphi(x,z)]$$

by Σ_1-induction,[4] from which

$$(\exists a)(\forall x < y)(\exists z < a)\varphi(x,z)]$$

follows by taking $t = y$.

- *base step* ($t = 0$)
  We want to prove

  $$0 \leq y \ \rightarrow \ (\exists a)(\forall x < 0)(\exists z < a)\varphi(x,z),$$

  which is trivial because there is no $x < 0$.
- *induction step* ($t + 1$)
  We want to prove

  $$t + 1 \leq y \ \rightarrow \ (\exists a)(\forall x < t)(\exists z < a)\varphi(x,z).$$

  By induction hypothesis,

  $$t \leq y \ \rightarrow \ (\exists a)(\forall x < t)(\exists z < a)\varphi(x,z).$$

  By the hypothesis on y,

  $$t < y \ \rightarrow \ (\exists z)\varphi(t,z).$$

  By putting together the bound for all $x < t$ (i.e. a) and the value for $x = t$ (i.e. z), one thus obtains a bound for all $x \leq t$, i.e. for all $x < t + 1$.

[4]Notice that the formula in square brackets is indeed Σ_1^0. Since we cannot use Σ_1-collection (which we have to prove) to claim closure under bounded quantification, the trick is to note that φ is of the form $\exists s\psi$ for some quantifier-free formula ψ and then bound both quantifiers $\exists z \exists s$ by a.

- *an $\mathcal{M}$-r.e. $\mathcal{M}$-bounded set is $\mathcal{M}$-finite*
  Given an $\mathcal{M}$-r.e. $\mathcal{M}$-bounded set C, let $c \in \mathcal{M}$ be a bound for it and let g be an $\mathcal{M}$-recursive function with range C. Define a partial $\mathcal{M}$-recursive function f with range C as follows. Start with $f(0) = g(0)$. For any stage $s > 0$, let $f(s)$ be the first element generated by g that is not yet in the range of f.

  Consider now the Σ_1^0-formula
  $$\varphi(x) \Leftrightarrow f(x)\downarrow \Leftrightarrow (\exists y)(f(x) = y).$$
  In principle, there are three possible cases:

  - If $\varphi(0)$ does not hold in $\mathcal{M}$, then C is empty.
  - If there is an x such that $\varphi(x)$ holds in $\mathcal{M}$ but $\varphi(x+1)$ does not, then there is a last element generated in C, namely $f(x)$.
  - Otherwise, $(\forall x)\varphi(x)$ holds in $\mathcal{M}$ because the latter is a model of Σ_1-induction. But since f is one-one by definition, this means that for every x there are x elements of C, and hence x elements below the fixed element c that bounds C, which is impossible.

  In practice thus the third case cannot happen. In the remaining two, C is $\mathcal{M}$-finite. $\square$

Notice that, while the first part of the proof only requires Σ_1-collection (and it actually amounts to prove that the latter follows from Σ_1-induction), the second part really requires Σ_1-induction. Indeed, *in any model $\mathcal{M}$ of Σ_1-collection in which Σ_1-induction fails, there is an $\mathcal{M}$-r.e. $\mathcal{M}$-bounded and $\mathcal{M}$-infinite set* (let φ be a Σ_1^0 formula for which induction fails in $\mathcal{M}$ and let C be the set of elements x such that $(\forall y \leq x)\varphi(y)$ holds in $\mathcal{M}$. First, C is $\mathcal{M}$-r.e. because it is defined on $\mathcal{M}$ by a Σ_1^0 formula (by Σ_1-collection, the Σ_1^0-formulas are closed under bounded quantification). Second, C is $\mathcal{M}$-infinite because it contains 0 and is closed under successor. Third, C it is $\mathcal{M}$-bounded by any element x such that $\varphi(x)$ fails in $\mathcal{M}$).

Mytilinaios [1989] has extended the previous proof (by methods patterned on those used for the same purpose on Σ_1-admissible ordinals) to obtain the analogues of all results in Section X.1. One could thus guess that Σ_1-induction is an appropriate abstract setting for the finite injury priority method. Groszek and Slaman [199?a] have proved that it is indeed so, in the sense that if Σ_1-induction fails, then there is some (artificial) finite injury priority argument that cannot be carried out. For natural arguments, however, Σ_1-induction is stronger than needed. For example, Σ_1-collection is sufficient to solve Post's Problem (Slaman and Woodin [1989]) and to prove the Friedberg-Muchnik

Theorem (Chong and Mourad [1992]). It is not known whether the same is true with even less than Σ_1-collection, e.g. with $\mathcal{PA}^-$ alone.

Σ_2-collection is sufficient to carry out some typical infinite injury arguments, such as a proof of the Density Theorem (Groszek, Mytilinaois and Slaman [1996]), but not all. For example, *there are models of* Σ_2*-collection in which every incomplete* $\mathcal{M}$*-r.e. set has low* $\mathcal{M}$*-degree* (Mytilinaios and Slaman [1988]), although in every model of Σ_2-induction there are sets of incomplete high $\mathcal{M}$-r.e. $\mathcal{M}$-degree (Grozsek and Mytilinaios [1990]). The models of Σ_2-collection in which there are sets of incomplete high $\mathcal{M}$-r.e. $\mathcal{M}$-degree have been completely characterized as those satisfying Σ_2-induction (Chong and Yang [1998]).

The lattice-theoretical properties of $\mathcal{M}$-r.e. sets can be defined in the usual way. It is known that: $\mathcal{M}$-simple sets exist in every model of Σ_1-induction (by the proof of X.4.5); $\mathcal{M}$-hyperhypersimple sets exist in every model of Σ_2-collection (Chong [1989a]); $\mathcal{M}$-maximal sets exist in every model of Σ_2-induction (Chong [1989b]). The models of Σ_2-collection in which there are $\mathcal{M}$-maximal sets have been completely characterized as those satisfying Σ_2-induction (Chong [1989], Chong and Yang [1998]).

The results just quoted show that the amount of Induction or Collection needed to formalize a proof is a good measure of the strength needed to carry out priority arguments and prove degree-theoretical results. The setting of restricted induction is thus more suitable for an axiomatic treatment of the priority method than admissibility was, although the former can be seen a reformulation of the latter in a different setting (basically, nonstandard models of $+$ and $\times$ as opposed to nonstandard models of $<$) and freely uses notions and techniques that were originally discovered or invented in the context of admissibility.

For background on bounded induction, see Hájek and Pudlák [1993]. For more on the present topic, see Chong [1989], [1989a], [1989b], [1992], Chong and Mourad [1990], [1992], Chong and Yang [1997], [1998], Grozsek and Hummel [199?], Groszek and Mytilinaios [1990], Groszek, Mytilinaios and Slaman [1996], Grozsek and Slaman [199?], [199?a], Hájek and Kučera [1989], Mourad [199?], [199?a], Mytilinaios [1989], Mytilinaios and Slaman [1988], Slaman and Woodin [1989], Yang [1995], [1995a].

Avoiding the priority method

A final question, after all this discussion on the priority method, is whether it was not possible to avoid it at all. The ironic answer is indeed positive, at least partially, as we now see. For example, it is possible to solve Post's Problem by a priority-free argument, which however lacks the flexibility needed to make it a complete replacement for the priority method.

To illustrate the point we again attempt to construct a nonrecursive T-incomplete r.e. set A, by satisfying the usual requirements:

$$\begin{array}{lcl} P_e & : & \mathcal{W}_e \text{ infinite } \Rightarrow \mathcal{W}_e \cap A \neq \emptyset \\ N_e & : & \mathcal{K} \not\simeq \{e\}^A. \end{array}$$

This time, however, we will avoid any conflict by satisfying them separately, by means of two different but compatible constructions.

The obvious idea is to satisfy, on the one hand, the negative requirements by building an auxiliary nonrecursive set $C <_T \mathcal{K}$ by finite extensions (V.2.3) and, on the other hand, the positive requirements by building a simple set $A \leq_T C$ by permitting (III.3.18). The problem is that permitting for the construction of r.e. sets does not always work below Δ_2^0 sets, for example not below 1-generic degrees (XI.2.3.3), nor below minimal degrees (XI.4.5). We thus have to find a nice class of Δ_2^0 sets, below which one can do permitting.

Proposition X.4.6 Permitting below fixed-point free degrees (Kučera [1986]) *Any function $f \leq_T \mathcal{K}$ without fixed-points, i.e. such that*

$$(\forall x)(\mathcal{W}_x \neq \mathcal{W}_{f(x)}),$$

bounds an r.e. nonrecursive set A.

Proof. Given $f \leq_T \mathcal{K}$ without fixed-points, we use the method of III.3.18 to build an r.e. nonrecursive set A such that $A \leq_T f$. At stage s, for every $e \leq s$ such that $\mathcal{W}_{e,s} \cap A_s = \emptyset$, we put in A the smallest x such that

$$x > 2e \ \wedge \ x \in \mathcal{W}_{e,s} \ \wedge \ x \text{ is permitted by } f \text{ at stage } s.$$

As usual, permitting means that x can enter A only if an approximation of f changes at something smaller than x. As in the proof of III.1.5, we can force such a change by taking an appropriate fixed-point and then using the fact that the true f has instead no fixed-point. We thus consider, by the Limit Lemma IV.1.17, a recursive function f_1 such that

$$f(x) = \lim_{s\to\infty} f_1(x,s),$$

and proceed as follows.

Fix e. We look for a stage s at which we can satisfy P_e, i.e. such that

$$x > 2e \ \wedge \ x \in \mathcal{W}_{e,s} \ \wedge \ x > z$$

for some $x \leq s$, and some z such that

$$\mathcal{W}_z = \mathcal{W}_{f_1(z,x)}.$$

Obviously, z is defined by the Fixed-Point Theorem, uniformly in e. We can thus suppose we have a recursive function g such that

$$\mathcal{W}_{g(e)} = \begin{cases} \mathcal{W}_{f_1(g(e),x)} & \text{for the smallest } s \text{ and } x \leq s \text{ as above, if they exist} \\ \emptyset & \text{otherwise.} \end{cases}$$

Since f has no fixed-points,

$$\mathcal{W}_{g(e)} \neq \mathcal{W}_{f(g(e))},$$

so the approximation $f_1(g(e), x)$ to $f(g(e))$ must be wrong, i.e. it must change at some stage after x. But if we want to use this for permitting, we must avoid the case that $f_1(g(e), x)$ has already changed at some stage between x and the stage at which we are actually putting x into A. This can easily be taken care of, by inserting in the construction the request that we put x in A at stage s only if $f_1(g(e), x)$ has not changed since stage x.

The construction is thus as follows. At stage s, for every $e \leq s$ such that $\mathcal{W}_{e,s} \cap A_s = \emptyset$, put into A the smallest $x \leq s$ such that

- $x > 2e \ \wedge \ x \in \mathcal{W}_{e,s} \ \wedge \ x > g(e)$
- $f_1(g(e), x) = f_1(g(e), t)$, for each t such that $x \leq t \leq s$

if one exists.

A is obviously coinfinite by the condition $x > 2e$, as for Post's simple set (III.2.11). Moreover, A is simple because each P_e is satisfied, as we now prove. Given e, let s_e be a stage after which $f_1(g(e), s_e)$ no longer changes. If $\mathcal{W}_e$ is infinite, there is x such that

$$x > 2e \ \wedge \ x \in \mathcal{W}_e \ \wedge \ x > g(e) \ \wedge \ x > s_e.$$

Notice that $f_1(g(e), x)$ no longer changes, because $x > s_e$. Then, as soon as x is generated in $\mathcal{W}_e$, either P_e has already been satisfied, or it can be satisfied then.

It remains to prove that $A \leq_T f$. First, notice that a given x can enter A only for the sake of some P_e, in which case $x > g(e)$. Second, x can enter at stage s only if $f_1(g(e), x)$ has not changed between x and s. Since $x > g(e)$, consider all possible $y \leq x$ and all possible $f_1(y, x)$. Compute (recursively in f) their final values $f(y)$ and find (recursively) a stage $s_x \geq x$ such that, for all $y \leq x$,

$$f_1(y, s_x) = f(y).$$

Then x cannot enter A at any stage $s \geq s_x$. If it did, by construction

$$f_1(g(e), x) = f_1(g(e), s_x) = f_1(g(e), s)$$

because $x \leq s_x \leq s$. Then

$$\mathcal{W}_{g(e)} = \mathcal{W}_{f_1(g(e),x)} = \mathcal{W}_{f_1(g(e),s_x)} = \mathcal{W}_{f(g(e))}$$

by definition of g and the choice of s_x, contradicting the fact that f has no fixed-points. Thus

$$x \in A \ \Leftrightarrow \ x \in A_{s_x},$$

and this decides whether x is in A, recursively in (s_x, and hence in) f. $\square$

Corollary X.4.7 A priority-free solution to Post's Problem (Kučera [1986]) *An r.e. set which is neither recursive nor T-complete can be constructed without any use of the priority method.*

Proof. The previous proof uses permitting and the Fixed-Point Theorem, but not the priority method. Since the same is true of III.1.6, in which the existence of a function $f <_T \mathcal{K}$ without fixed-points is proved, we thus obtain a solution to Post's Problem without using the priority method. $\square$

Hájek and Kučera [1989] have proved that the previous argument still requires Σ_1-induction to be carried out. From the point of view of bounded induction, although only from it, there is thus no improvement over the usual proofs by the priority method.

As a bonus, we can also obtain a proof of a result that we only quoted after V.5.39.

Corollary X.4.8 (Kučera [1986]) *There is no complete extension of $\mathcal{PA}$ in which only Δ_2^0 sets are weakly representable, but no nonrecursive r.e. set is.*

Proof. The sets weakly representable in a complete extension $\mathcal{F}$ of $\mathcal{PA}$ form a basis for Π_1^0-classes, by an argument as in V.5.35. Thus there is such a set B of the same degree as a function f without fixed-points (by an argument as in III.1.6). If only Δ_2^0 sets are weakly representable in $\mathcal{F}$, then $f \leq_T \mathcal{K}$. By X.4.6, there is an r.e. nonrecursive set $A \leq_T f \leq_T B$. But the sets weakly representable in $\mathcal{F}$ are closed downward w.r.t. $\leq_T$. Thus A is representable in $\mathcal{F}$ too. $\square$

Variations of the same method to prove various other results, including the Friedberg-Muchnik Theorem, are in Kučera [1986], [1988], [1989]. Although one can also obtain alternative, priority-free proofs of some results originally proved by infinite injury, such as the existence of minimal pairs of r.e. degrees (X.6.5) or Sacks' Jump Inversion Theorem (XI.1.23), these proofs are not as neat as the previous solution to Post's Problem. On the one hand, the degrees without fixed-points cannot in general be below $\mathbf{0}'$, since a modification of the proof of X.4.6 shows that any two such degrees bound a nonrecursive r.e. degree (thus, for example, they cannot be a minimal pair in the degrees below $\mathbf{0}'$). On the other hand, degrees below $\mathbf{0}''$ without fixed-points do not necessarily bound nonrecursive r.e. degrees (because, by V.5.34, there are such degrees that are hyperimmune-free and hence, by V.5.3.d, not above $\mathbf{0}'$).

A review of solutions to Post's Problem $\star$

By now we have accumulated a good number of different solutions to Post's Problem. It is now time to pause for a moment to discuss their relevant features.

The solution provided in III.5.20 was along the lines originally proposed by Post [1944] and pursued in Sections III.2–5, namely to find *structural properties of the complement of a set* that would guarantee nonrecursiveness and incompleteness. Marchenkov [1976] proved that being semirecursive and η-hyperhypersimple is sufficient to guarantee incompleteness, and the notion of η-hyperhypersimplicity is a variation of the notion of hyperhypersimplicity introduced by Post himself.

The requirement of working only with *lattice-theoretical properties definable in $\mathcal{E}$* was certainly not a condition that Post considered (essential) for the kind of solution he was pursuing. On the one hand, he himself considered the notion of hypersimplicity, which is not definable (IX.3.24). On the other hand, hyperhypersimplicity turned out to be definable only in a surprising and nontrivial way (IX.2.10). But the attempt of solving Post's Problem by means of a property definable in $\mathcal{E}$ is certainly a natural extension of Post's original program. On the negative side, Cholak, Downey and Stob [1992] have proved that no definable property of $\overline{A}$ alone can guarantee incompleteness of A, while Cholak [1995] and Harrington and Soare [199?b] have proved that no definable property of A can imply that the degree of A is not high. On the positive side, Harrington and Soare [1991], [1996] have found a property of the following kind:

> A is a major subset of a set B and if $C \supseteq \overline{B}$ is r.e., then the elements that first enter C cannot promptly enter A

that does guarantee incompleteness of A (since A is a major subset, it must then have incomplete high degree).

The original solution to Post's Problem was provided by Muchnik [1956] and Friedberg [1957a], who independently proved by *direct constructions* that there are incomparable r.e. degrees (X.1.7). Obviously, all nontrivial results about r.e. degrees, in particular all those proved in Sections 1 and 6, provide alternative solutions to Post's Problem.

Two solutions of a different kind have been obtained by Rogers [1959], by considering *index sets* of classes of r.e. sets. The first follows from the fact that the index sets of the nonrecursive and of the T-incomplete r.e. sets are different.

The second is obtained by the Fixed-Point Theorem, using a characterization of the complexity of the index sets of the recursive sets and of the T-complete r.e. sets (see p. 632).

Structural solutions to Post's Problem defining, without any mention of T-reducibility, *degree-theoretical properties* that imply nonrecursiveness and T-incompleteness, such as lowness (XI.1.9) or cappability (X.6.9), have been proposed by Soare [1977] and Ambos-Spies and Nies [1992].

All results quoted above, from the existence of nonrecursive, semirecursive, η-hyperhypersimple sets to the existence of cappable degrees, are proved by the priority method. The method of *permitting below fixed-point free degrees*, introduced by Kučera [1986], produces instead a solution to Post's Problem without any use of the priority method (X.4.6).

Despite their great abundance, all known solutions to Post's Problem are quite unnatural, since the r.e. sets they exhibit are especially constructed for the purpose. A number of proposals for more satisfactory solutions to the problem have thus been advanced.

Sacks [1966] has asked for a *degree invariant solution*, i.e. an r.e. set $\mathcal{W}_e$ such that

$$(\forall X)(\forall Y)(X \equiv_T Y \Rightarrow \mathcal{W}_e^X \equiv_T \mathcal{W}_e^Y) \wedge (\forall X)(X <_T \mathcal{W}_e^X <_T X'),$$

and the following is known:

- Lachlan [1975b] has proved that there is no uniformly degree invariant solution, i.e. one in which the first condition is strengthened by requiring that indices of the two reductions in $\mathcal{W}_e^X \equiv_T \mathcal{W}_e^Y$ can be obtained recursively from indices of the two reductions in $X \equiv_T Y$. Such uniformly degree invariant r.e. sets obviously exist, e.g. any e such that $\mathcal{W}_e^X = X$ or $\mathcal{W}_e^X = X'$, but they cannot be solutions to Post's Problem relative to X.

- Downey and Shore [1997] have proved that there is no degree invariant solution in which the second condition is strengthened by requiring that $\mathcal{W}_e^X$ is always of intermediate degree, i.e. such that

$$(\forall X)(\forall n)(X^{(n)} <_T (\mathcal{W}_e^X)^{(n)} <_T X^{(n+1)}).$$

  Such solutions to Post's Problem exist, by the proof of XI.1.19, but they cannot be degree invariant.

- Despite these negative obstructions, Slaman has obtained a partial positive solution that is degree invariant for all $X \leq_T \mathcal{K}$, where $\mathcal{K}$ is Σ_1^0-complete. A full positive solution would require an extension to all $X \leq_T \mathcal{O}$, where $\mathcal{O}$ is Π_1^1-complete (because if $\mathcal{W}_e^X$ is not a degree invariant solution to Post's Problem, then the counterexamples form a nonempty Σ_1^1 class, which contains a member recursive in $\mathcal{O}$ by the Kleene Basis Theorem).

A weaker formulation of a degree invariant solution to Post's Problem asks for a definable function $F : \mathcal{P}(\omega) \to \mathcal{P}(\omega)$ well-defined on degrees, inducing a function $f : \boldsymbol{\mathcal{D}} \to \boldsymbol{\mathcal{D}}$ such that

$$(\forall \boldsymbol{a})(f(\boldsymbol{a}) \in \boldsymbol{\mathcal{R}}^{\boldsymbol{a}}) \ \wedge \ (\forall \boldsymbol{a})(\boldsymbol{a} < f(\boldsymbol{a}) < \boldsymbol{a}'),$$

where $\boldsymbol{\mathcal{R}}^{\boldsymbol{a}}$ is the set of degrees r.e. in and above $\boldsymbol{a}$. Steel [1982] has proved that no such function can be uniform, thus extending Lachlan's result quoted above (which holds for the Σ_1^0 functions defined by $F(X) = \mathcal{W}_e^X$, for some e).

Slaman has asked for a solution to Post's Problem *definable in* $\mathcal{R}$, i.e. for a nontrivial definable r.e. degree. But the most satisfactory solution to the problem would be the discovery that *some r.e. set naturally occurring in mathematics*, e.g. in the form of the set of theorems of an interesting formal system, turns out to be nonrecursive and T-incomplete.

X.5 Many-One Degrees

In this section we deal with r.e. **$\boldsymbol{m}$-degrees**, and leave the study of other strong reducibilities for Section 8. We have already discussed at the beginning of the chapter the reasons why we think that m-degrees are the natural setting for the study of r.e. sets. We add here the fact that *$\boldsymbol{\mathcal{R}_m}$ is an ideal of $\boldsymbol{\mathcal{D}_m}$*, unlike what happens for any other weaker reducibility (since there are two r.e. sets A and B such that the Turing degree of $A - B$ does not contain any r.e. set, see XI.5.9.a, and $A - B \leq_{btt} \mathcal{K}$).

Incomparable degrees

We already know that there are incomparable r.e. m-degrees, since there are incomparable r.e. Turing degrees.

Sacks agreement method showed us how to avoid upper cones. We now introduce a method to **avoid lower cones**, and by combining the two we are able to produce degrees incomparable with a given nontrivial one. The ideas will be generalized to the more complicated situation of Turing degrees, where they will eventually produce the Density Theorem (X.6.3).

Proposition X.5.1 (Sacks [1963a], Denisov [1970]) *For every r.e. m-degree $\boldsymbol{c}$ such that $\boldsymbol{0_m} < \boldsymbol{c} < \boldsymbol{0'_m}$, there is an incomparable r.e. m-degree $\boldsymbol{a}$.*

Proof. Let C be an r.e. nonrecursive set such that $\mathcal{K} \not\leq_m C$. We want to build an r.e. set A such that

$$\begin{array}{rcl} P_e & : & A \not\leq_m C \text{ via } \varphi_e \\ N_e & : & C \not\leq_m A \text{ via } \varphi_e. \end{array}$$

To satisfy N_e we use Sacks agreement method. In this context we have

$$\begin{array}{rcl} u(e,x,s) & = & \begin{cases} \varphi_{e,s}(x) & \text{if convergent} \\ 0 & \text{otherwise} \end{cases} \\ l_n(e,s) & = & \max\{z : (\forall y < z)[y \in C_s \Leftrightarrow \varphi_{e,s}(y) \in A_s]\} \\ r(e,s) & = & \max\{u(e,x,s) : x \leq l_n(e,s)\} \\ & = & \max\{\varphi_{e,s}(x) : \varphi_{e,s}(x)\downarrow \wedge\, x \leq l_n(e,s)\}, \end{array}$$

where l_n measures the length of agreement for negative requirements.

To satisfy P_e we need a new strategy (the **coding method**) dual to that above. In the words of Sacks [1964a], we keep planting members of $\mathcal{K}$ into A until A looks enough like $\mathcal{K}$ to guarantee that A is not m-reducible to C. We thus consider the e-th column of A, and try to code $\mathcal{K}$ into it. But we cannot do this without restrain, since otherwise A would become m-complete. Thus we code $\mathcal{K}$ into A only as long as it seems that $A \leq_m C$ via φ_e. If at the end we really had $A \leq_m C$ via φ_e, then (except for finitely many elements, due to restraints of higher priority) we would have

$$x \in \mathcal{K} \Leftrightarrow \langle e, x\rangle \in A,$$

i.e. $\mathcal{K} \leq_m A \leq_m C$, which is a contradiction. The following function measures the agreement method for positive requirements:

$$l_p(e,s) = \max\{z : (\forall y < z)[y \in A_s \Leftrightarrow \varphi_{e,s}(y) \in C_s]\}.$$

The construction is as follows. We start with $A_0 = \emptyset$. At stage $s+1$, we put $\langle e, x\rangle$ in A if

$$x \in \mathcal{K}_s \wedge (\forall i \leq e)[\langle e, x\rangle > r(i,s)] \wedge x < l_p(e,s).$$

- *N_e is satisfied, for every e*
  As in X.1.10, since C is nonrecursive and N_e is injured only finitely many times (if P_i is satisfied, then $\lim_{s\to\infty} l_p(i,s) < \infty$).

- $\lim_{s\to\infty} r(e,s) < \infty$*, for every e*
  As in X.1.10.

- *P_e is satisfied, for every e*
  Let s_0 be a stage large enough so that after it no N_i with $i \leq e$ is injured,

and $r(i, s_0)$ has permanently attained its limit for every $i \leq e$. If $A \leq_m C$ via φ_e, from $\lim_{s\to\infty} l_p(e, s) = \infty$ we obtain, for $\langle e, x\rangle > \max_{i\leq e} r(i, s_0)$,

$$x \in \mathcal{K} \Leftrightarrow \langle e, x\rangle \in A,$$

and so $\mathcal{K} \leq_m A \leq_m C$, which is a contradiction. $\square$

Upward density

With a few changes in the previous proof, we can obtain the following result.

Proposition X.5.2 (Sacks [1963a], Denisov [1970]) *Given an r.e. m-degree $\mathbf{c}$ such that $\mathbf{c} < \mathbf{0}'_{\boldsymbol{m}}$, there is an r.e. m-degree $\boldsymbol{a}$ such that $\mathbf{c} < \boldsymbol{a} < \mathbf{0}'_{\boldsymbol{m}}$.*

Proof. Let C be an r.e. set such that $C <_m \mathcal{K}$. We want to build an r.e. set A such that

$$\begin{array}{lll} P_{-1} & : & C \leq_m A \\ P_e & : & A \not\leq_m C \text{ via } \varphi_e \\ N_e & : & \mathcal{K} \not\leq_m A \text{ via } \varphi_e. \end{array}$$

P_e and N_e are handled as in X.5.1, respectively by the coding and the Sacks agreement methods. The only formal change is that, to satisfy P_e, we now try to code $\mathcal{K}$ into $A^{[e+1]}$ instead of $A^{[e]}$. So $A^{[0]}$ is free, and we use it to code C (to satisfy P_{-1}, with highest priority).

The construction is as follows. We start with $A_0 = \emptyset$. At stage $s + 1$, we put $\langle 0, x\rangle$ in A if $x \in C_s$, and we put $\langle e + 1, x\rangle$ in A if

$$x \in \mathcal{K}_s \wedge (\forall i \leq e)[\langle e + 1, x\rangle > r(i, s)] \wedge x < l_p(e, s),$$

where $r(e, s)$ is defined as usual, using

$$l_n(e, s) = \max \{z : (\forall y < z)[y \in \mathcal{K}_s \Leftrightarrow \varphi_{e,s}(y) \in A_s]\}.$$

The proof that the requirements are satisfied is as in X.5.1. Only, N_e is satisfied otherwise $\mathcal{K}$ would be reducible to C (since $A^{[0]} \equiv_m C$), contradicting the hypothesis.

For example, we show that N_0 is satisfied. Suppose $\mathcal{K} \leq_m A$ via φ_0. We want to show that $A \leq_m C$, so that $\mathcal{K} \leq_m C$, which is a contradiction. We define a recursive function f such that

$$x \in A \Leftrightarrow f(x) \in C.$$

If x is in the 0-th column, $x = \langle 0, z\rangle$ for some z. Let $f(x) = z$. Then $x \in A$ if and only if $f(x) \in C$.

If x is not in the 0-th column, from $\mathcal{K} \leq_m A$ via φ_0 we have

$$\lim_{s\to\infty} l_n(0,s) = \infty \qquad \text{and so} \qquad \lim_{s\to\infty} r(0,s) = \infty$$

(otherwise the range of φ_0 would be finite, and $\mathcal{K}$ would be recursive). Also, l_n never drops back (otherwise, since N_0 has highest priority, a disagreement would be preserved) and hence neither does r. Given x, we search for a stage s such that $r(0,s) > x$. Then $x \in A$ if and only if $x \in A_s$ and we can let $f(x)$ be any element of C if $x \in A_s$, and any element of $\overline{C}$ otherwise. $\square$

A different proof of X.5.2 follows from X.5.1 and X.5.9. More precisely, given $\mathbf{0_m} < \boldsymbol{c} < \mathbf{0'_m}$ one can first take $\boldsymbol{a}$ incomparable with it, and then consider $\boldsymbol{a} \cup \boldsymbol{b}$, which is strictly above $\boldsymbol{a}$ (because $\boldsymbol{a}$ and $\boldsymbol{b}$ are incomparable) and different from $\mathbf{0}'_m$ (because no pair of nontrivial m-degrees joins to $\mathbf{0}'_m$).

Exercise X.5.3 a) *Every countable partial ordering is embeddable in the r.e. m-degrees above any incomplete r.e. m-degree.* (Kallibekov [1973]) (Hint: use the methods of X.5.2 to build an r.e. set A such that $C \leq_m A$, and whose columns are independent w.r.t. m-reducibility.)

What keeps us from obtaining full density (i.e., given r.e. m-degrees $\boldsymbol{b} < \boldsymbol{c}$, to find an r.e. m-degree $\boldsymbol{a}$ such that $\boldsymbol{b} < \boldsymbol{a} < \boldsymbol{c}$) is that we do not know how to push an m-degree below a given nonrecursive one. We will see in X.5.11 that density fails for r.e. m-degrees (as well as for r.e. *btt*-degrees and *tt*-degrees, see X.7.8 and X.7.16). But for *wtt*-reducibility, where permitting holds (p. I.338), the same proof of X.5.2 shows that *the r.e. wtt-degrees are dense* (Ladner and Sasso [1975]). The same result holds for r.e. Turing degrees (X.6.3), with a more complicated proof (extending the coding method of X.5.2).

Greatest lower bounds

We now consider the existence of g.l.b.'s for pairs of r.e. m-degrees, and prove that they exist sometimes, but not always.

The next result is a trivial consequence of the existence of two minimal r.e. m-degrees, but the direct proof given below extends (with appropriate modifications) to r.e. Turing degrees (X.6.5).

Proposition X.5.4 (Young [1963]) *There exists a minimal pair of r.e. m-degrees.*

Proof. We want to build two r.e. sets A and B such that

$$\begin{array}{lll} P_{2e} & : & \mathcal{W}_e \text{ infinite } \Rightarrow \mathcal{W}_e \cap A \neq \emptyset \\ P_{2e+1} & : & \mathcal{W}_e \text{ infinite } \Rightarrow \mathcal{W}_e \cap B \neq \emptyset \\ N_e & : & C \leq_m A, B \text{ via } \varphi_{(e)_1}, \varphi_{(e)_2} \Rightarrow C \text{ recursive.} \end{array}$$

The idea is to ensure the satisfaction of N_e is as follows. Whenever we discover that

$$\varphi_{(e)_1}(x) \in A_s \;\Leftrightarrow\; \varphi_{(e)_2}(x) \in B_s,$$

there are two possibilities:

- Both numbers are in the respective sets, so the situation is permanent (because the construction never takes an element out of A or B, after it has gone in).

- Both numbers are out of the respective sets. We then decide to put at most one in, with the following effect. Either one is put in and the other is not, and then we have disagreement and N_e is vacuously satisfied, because

  $$\varphi_{(e)_1}(x) \in A \;\Leftrightarrow\; \varphi_{(e)_2}(x) \notin B.$$

  Or none of them is put in, and then the computation is final. If the latter case happens for every x, then N_e is satisfied because, since

  $$x \in C \;\Leftrightarrow\; \varphi_{(e)_1}(x) \in A \;\Leftrightarrow\; \varphi_{(e)_2}(x) \in B,$$

  C is recursive. Indeed, to compute $C(x)$ it is enough to find an s such that $\varphi_{(e)_1,s}(x)$ converges, and see whether $\varphi_{(e)_1}(x) \in A_s$.

In other words, we just rule out the only troublesome case, namely to let the agreement first be ruined, and then restored with a different value (so that we would not know, at any given stage, whether the situation is permanent).

We now analyze more carefully what happens when injuries are allowed. First let

$$l(e,s) \;=\; \max\{z : (\forall y < z)[\varphi_{(e)_1,s}(y) \in A_s \;\Leftrightarrow\; \varphi_{(e)_2,s}(y) \in B_s]\}.$$

Call s an ***e*-maximal stage** if $l(e,s) \geq \max_{t<s}\ l(e,t)$. Until we only have e-maximal stages, we do not have to impose any restraint because we know that we can destroy one side or the other, as long as we do not destroy both. We thus let $r(e,s) = 0$.

Suppose s_0 is the first stage that is not e-maximal. Then

$$l(e,s_0) < l(e,s_0-1) = \max_{t<s_0}\ l(e,t).$$

This means that we have destroyed one side, and we now want to avoid a destruction of the other. We let, for simplicity, $r(e,s_0) = s_0 - 1$. Since every computation that converges at stage s must use only elements less than s, a restraint with value s will protect every convergent computation $\varphi_{i,s}(y)$. Thus, by setting $r(e,s_0) = s_0 - 1$, we protect every computation below $l(e,s_0-1)$. If we succeed (i.e. there is no later injury), then N_e is vacuously satisfied. Suppose instead there are injuries. We might restore some agreement, and $l(e, s)$ could increase. But as long as $l(e, s)$ remains below $l(e, s_0 - 1)$, we can continue to let $r(e, s) = s_0 - 1$, since by protecting this we still ensure some disagreement.

If at a certain point we have $l(e, s) \geq l(e, s_0 - 1)$, then the initial length of agreement has been restored, and we can begin again with our initial strategy, since if we succeed from this stage on in preserving one side, we can still prove that C is recursive.

With this motivation in mind, we define the restraint function as

$$r(e,s) = \begin{cases} 0 & \text{if } s \text{ is } e\text{-maximal} \\ \text{the greatest } e\text{-maximal stage } t < s & \text{otherwise.} \end{cases}$$

The construction is as follows. We start with $A_0 = B_0 = \emptyset$. At stage $s+1$, consider the least positive requirement with index $\leq s$ that has not yet been satisfied and can be satisfied now, and then satisfy it. More precisely, let $e \leq s$ be the smallest number such that:

- $\mathcal{W}_{e,s} \cap A_s = \emptyset$ or $\mathcal{W}_{e,s} \cap B_s = \emptyset$
- for some x, $x \in \mathcal{W}_{e,s} \ \wedge \ x \geq 2e \ \wedge \ (\forall i \leq e)[x > r(i, s)]$

and put one such x in A if $\mathcal{W}_{e,s} \cap A_s = \emptyset$, and in B otherwise.

Note that it is crucial for the construction that at most one positive requirement be satisfied at every stage, because this ensures that at most one side is injured.

- *N_e is satisfied, for every e*
  Since N_e can be injured only finitely many times by positive requirements of higher priority, let s_0 be a stage after which N_e is not injured. Suppose that
  $$x \in C \ \Leftrightarrow \ \varphi_{(e)_1}(x) \in A \ \Leftrightarrow \ \varphi_{(e)_2}(x) \in B$$
  for every x. C is recursive because it is enough, given x, to find $s > s_0$ such that $l(e, s) > x$ and s is e-maximal (s exists because $\lim_{s\to\infty} l(e, s) = \infty$, hence there are infinitely many e-maximal stages). Then
  $$x \in C \ \Leftrightarrow \ \varphi_{(e)_1}(x) \in A_s,$$
  otherwise disagreement would be preserved forever between $\varphi_{(e)_1}(x) \in A_s$ and $\varphi_{(e)_2}(x) \in B_s$.

- $\lim_{s\to\infty} r(e, s) < \infty$, *for every e*
  Let s_0 be as above. If
  $$\varphi_{(e)_1}(x) \in A \ \Leftrightarrow \ \varphi_{(e)_2}(x) \in B$$

for every x, then what we proved above shows that every stage $s \geq s_0$ is e-maximal, and $\lim_{s\to\infty} r(e,s) = 0$. If

$$\varphi_{(e)_1}(x) \in A \Leftrightarrow \varphi_{(e)_2}(x) \notin B$$

for some x, choose $s_1 \geq s_0$ large enough so that

$$\begin{array}{lcl} \varphi_{(e)_1,s_1}(x) \in A_{s_1} & \Leftrightarrow & \varphi_{(e)_1}(x) \in A \\ \varphi_{(e)_2,s_1}(x) \in B_{s_1} & \Leftrightarrow & \varphi_{(e)_2}(x) \in B. \end{array}$$

Then, for every $s \geq s_1$, $l(e,s) \leq l(e,s_1)$ and again $\lim_{s\to\infty} r(e,s)$ exists.

- *P_e is satisfied, for every e*
  As usual. □

The simple proof above fails for *wtt* and T-degrees, since the length of agreement may drop in these cases because one or both sides becomes undefined, and the restraint would only freeze convergent computations. The result still holds for both *wtt* and T-degrees, but with a more complicated proof (see X.6.5).

In X.5.22 we will prove a strong generalization of X.5.4, to the effect that every nontrivial r.e. m-degree is part of a minimal pair. This fails for r.e. Turing degrees (X.6.24).

Exercise X.5.5 Branching r.e. m-degrees. An r.e. m-degree is called **branching** if it is the g.l.b. of two incomparable r.e. m-degrees above it.

Branching Theorem for $\mathcal{R}_m$. *Every incomplete r.e. m-degree is branching.* (Hint: given an r.e. set D such that $\mathcal{K} \not\leq_m D$, build two r.e. sets A and B such that $D \leq_T A, B$ and satisfying the requirements

$$\begin{array}{lcl} P_{2e} & : & A \not\leq_m D \text{ via } \varphi_e \\ P_{2e+1} & : & B \not\leq_m D \text{ via } \varphi_e \\ N_e & : & C \leq_m A, B \text{ via } \varphi_{(e)_1}, \varphi_{(e)_2} \Rightarrow C \leq_m D, \end{array}$$

by combining the proofs of X.5.2 and X.5.4.)

We turn now to pairs without g.l.b.

Proposition X.5.6 (Ershov [1969], Lachlan [1972a]) *There are two r.e. m-degrees without g.l.b.*

Proof. As in V.4.7, we want to build an infinite ascending sequence of r.e. m-degrees and an r.e. exact pair for it. We build A such that the chain of the m-degrees of its sections $A^{[<e]}$ is strictly increasing, and a thick subset B of A

(so that its sections have the same m-degrees as the corresponding sections of A) such that A and B form an exact pair for the chain.

The requirements are:

$$\begin{array}{lll} R^A_{e,n} & : & A^{[e]} \not\leq_m A^{[<e]} \text{ via } \varphi_n \\ P^B_e & : & B^{[e]} =^* A^{[e]} \\ N_e & : & C \leq_m A, B \text{ via } \varphi_{(e)_1}, \varphi_{(e)_2} \ \Rightarrow\ C \leq_m A^{[n]}, \text{ for some } n. \end{array}$$

Notice that it is not necessary to force $B \not\leq_m A$, since this is automatically ensured by the other requirements: if $B \leq_m A$, then N-requirements (with $C = B$) would give $B \leq_m A^{[n]}$ for some n, the P-requirements $B \leq_m B^{[n]}$, and the R-requirements $B \not\leq_m B^{[<n+1]}$, contradiction.

N_e is treated as in the minimal pair construction, and imposes (on both A and B) a negative restraint $r(e,s)$ defined as in X.5.4. Also, it does not allow us to put elements in both A and B at the same stage.

To satisfy P^B_e we put in B every element which has gone into $A^{[e]}$ and is not restrained, i.e. when P^B_e has highest priority we ensure that if

$$x \in A^{[e]}_s - B_s \ \wedge\ (\forall i \leq e)[x > r(i,s)],$$

then $x \in B_{s+1}$.

To satisfy $R^A_{e,n}$ we unfold the e-th column, so that we choose witnesses for it in the n-th column of the e-th column. The strategy is the usual one of taking witnesses $x \notin A_s$, wait until $\varphi_n(x)$ converges, and then ensure

$$x \in A^{[e]} \ \Leftrightarrow\ \varphi_n(x) \notin A^{[<e]}.$$

We make sure that, at each stage, the current witness is greater than $r(i,s)$ for $i \leq e$, and not restrained by conditions $R^A_{i,m}$ of higher priority.

We leave to the reader the task of arranging the requirements in a priority list, spelling out the construction, and checking that it works. □

Corollary X.5.7 $\boldsymbol{\mathcal{R}_m}$ *is not a lattice.*

Least upper bounds

The behavior of l.u.b.'s for r.e. m-degrees is opposite to that of g.l.b.'s.

On the one hand, we already know (VI.1.1) that $A \oplus B$ is the l.u.b. of A and B w.r.t. m-reducibility, and it is obviously r.e. if both A and B are. Thus *every pair of r.e. m-degrees $\boldsymbol{a}$ and $\boldsymbol{b}$ has an r.e. l.u.b. $\boldsymbol{a} \cup \boldsymbol{b}$*, and the analogue of X.5.6 fails.

On the other hand, the analogue of X.5.4 also fails.

Proposition X.5.8 Non-Cupping Theorem for $\mathcal{R}_m$ (Lachlan [1966]) *If $\boldsymbol{a}$ and $\boldsymbol{b}$ are less than $\mathbf{0}'_m$, so is $\boldsymbol{a} \cup \boldsymbol{b}$. Thus no r.e. m-degree can be cupped to $\mathbf{0}'_m$.*

Proof. Let $A \oplus B \equiv_m \mathcal{K}$. Since $A \oplus B \leq_m A \cdot B$ for $A, B \neq \emptyset$, $A \cdot B$ is m-complete. Then so is one of A and B, by III.9.3. $\square$

Corollary X.5.9 Non-Splitting Theorem for $\mathcal{R}_m$ (Lachlan [1966]) $\mathbf{0}'_m$ *cannot be nontrivially split.*

Connections between structure and degrees ⋆

The fact that the notion of m-degree is natural for the study of r.e. sets is witnessed by the results of this subsection, that will expose a deep relationship between structural and degree-theoretical properties of r.e. sets.

We start with the following general observation, which extends VI.2.1.

Proposition X.5.10 (Lachlan [1972a]) *Let A be an r.e. set and $\boldsymbol{a}$ its m-degree. Then there is a homomorphism of uppersemilattices from $\mathcal{L}^*(A)$ onto $\mathcal{D}_m(\leq \boldsymbol{a})$.*

Proof. Given any r.e. set B such that $B \supseteq A$, let f be any recursive function with range B, and define

$$x \in \Phi(B) \Leftrightarrow f(x) \in A.$$

As in VI.2.1, one proves:

1. the m-degree of $\Phi(B)$ does not depend on f
2. $\Phi(B) \leq_m A$
3. if B and C are r.e. sets and $A \subseteq B \subseteq C$, then $\Phi(B) \leq_m \Phi(C)$
4. if B and C are r.e. supersets of A differing finitely, then $\Phi(B) \equiv_m \Phi(C)$
5. if $C \leq_m A$, then there is an r.e. set $B \supseteq A$ such that $\Phi(B) \equiv_m C$ (if f m-reduces C to A, let B be the union of A and the range of f). $\square$

We turn now to specific relationships between structure and m-degrees.

Proposition X.5.11 (Young [1963], McLaughlin [1965a]) *Maximal sets have minimal m-degrees.*

Proof. Let A be maximal, and D be an r.e. set such that $D \leq_m A$. Given a recursive function f such that

$$x \in D \Leftrightarrow f(x) \in A,$$

consider its range S.

If $f(\overline{D})$ is finite, then D is recursive, since $x \in \overline{D}$ if and only if $f(x) \in f(\overline{D})$.

If $f(\overline{D}) = S \cap \overline{A}$ is infinite, $\overline{S} \cap \overline{A}$ is finite by maximality. We want to show that $A \leq_m D$. Fix $a \in D$ and $b \in \overline{D}$, and simultaneously enumerate S and A. Let

$$g(z) = \begin{cases} \mu x.\,(f(x) = z) & \text{if } z \text{ shows up first in } S \\ a & \text{if } z \text{ shows up first in } A \\ b & \text{if } z \in \overline{S} \cap \overline{A}. \end{cases}$$

g is clearly recursive, since $\overline{S} \cap \overline{A}$ is finite. Moreover, in the first case

$$z \in A \Leftrightarrow f(x) \in A \Leftrightarrow x \in D \Leftrightarrow g(z) \in D.$$

In the last two cases,

$$z \in A \Leftrightarrow g(z) \in D$$

by definition. Thus $A \leq_m D$. $\square$

Note that a minimal r.e. m-degree is automatically a minimal m-degree, since the r.e. m-degrees are an ideal of the m-degrees. In VI.2.2 we built minimal m-degrees by a different method, closer to the following useful generalization of the last result.

Proposition X.5.12 (Ershov [1971], Lachlan [1972]) *η-maximal nonrecursive sets have minimal m-degrees.*

Proof. We proceed as above. If $[f(\overline{D})]_\eta$ is η-finite, then D is recursive because $x \in \overline{D}$ if and only $f(x)$ is η-equivalent to one of finitely many elements.

Otherwise, $[\overline{S} \cap \overline{A}]_\eta$ is η-finite. With notation as above, let

$$g(z) = \begin{cases} \mu x.\,(f(x) = z) & \text{if } z \text{ shows up first in } [S]_\eta \\ a & \text{if } z \text{ shows up first in } A \\ b & \text{if } z \in [\overline{S} \cap \overline{A}]_\eta. \end{cases}$$

Again g is recursive, because either $z \in [\overline{S} \cap \overline{A}]_\eta$ or $z \in A \cup [S]_\eta$. Moreover, g m-reduces A to D. $\square$

Exercises X.5.13 a) *Not every minimal r.e. m-degree contains a simple (and hence a maximal) set.* (Ershov [1970a], Degtev [1976]) (Hint: split a maximal set A into two disjoint and recursively inseparable r.e. sets B and C, see IX.2.5. The m-degree of B is minimal as in X.5.11, by adding the condition 'or z shows up first in C' in the last clause of the definition of g. If h m-reduced B to a simple set B_0, then $h(C)$ would be finite, and the set of z's such that $h(z) \in h(C)$ would recursively separate B and C.)

b) *The minimal m-degree of the proof of part a) contains η-maximal sets.* (Hint: identify x and y if they both belong to B, or to C.)

Degtev [1976] shows that *not every minimal r.e. m-degree contains an η-maximal set.*

In the case of hyperhypersimple sets, there is a particularly simple way of mapping $\mathcal{L}^*(A)$ onto $\boldsymbol{\mathcal{D}_m(\leq a)}$, by just sending a set to its m-degree.

Proposition X.5.14 (Ershov [1969]) *If A is hyperhypersimple and $\boldsymbol{a}$ is its m-degree, $\boldsymbol{\mathcal{D}_m(\leq a)}$ consists exactly of the m-degrees containing r.e. supersets of A.*

Proof. Let B be an r.e. superset of A. By IX.2.10, $A \cup \overline{B}$ is r.e. So, if $a \in A$,

$$g(x) = \begin{cases} a & \text{if } x \text{ shows up first in } B \\ x & \text{if } x \text{ shows up first in } A \cup \overline{B} \end{cases}$$

is total and recursive, and $B \leq_m A$ via g.

Conversely, suppose $C \leq_m A$ via f. If $B = A \cup \text{range } f$, $A \cup \overline{B}$ is r.e. by IX.2.10. It is enough to show that $C \equiv_m A \cup \overline{B}$. But $C \leq_m A \cup \overline{B}$ via f itself, and $A \cup \overline{B} \leq_m C$ via h defined as

$$h(x) = \begin{cases} \mu z.\,(f(z) = x) & \text{if } x \text{ shows up first in the range of } f \\ c & \text{if } x \text{ shows up first in } A \cup \overline{B}, \end{cases}$$

where $c \in C$. $\Box$

Exercises X.5.15 a) *The m-degrees of hyperhypersimple sets together with $\mathbf{0}_m$ form an ideal of the r.e. m-degrees.* (Ershov [1969]) (Hint: for downward closure, use X.5.14.)

b) *If A is hyperhypersimple, and B and C are r.e. sets such that $A \subseteq B \subseteq C$, then $C \leq_m B$.* Thus the map of X.5.14 is an antihomomorphism from $\mathcal{L}^*(A)$ onto $\boldsymbol{\mathcal{D}(\leq a)}$. (Ershov [1969]) (Hint: $A \cup \overline{C}$ is r.e.)

c) *If A is not simple, then every r.e. supserset of A is m-reducible to it if and only if A is either creative or cofinite.* (Degtev [1978]) (Hint: if A is coinfinite and not simple, there is an infinite recursive set $R \subseteq \overline{A}$. Let f be a recursive one-one function with range R, and $\mathcal{W}_{g(x)} = f(\mathcal{W}_x)$. Then $\mathcal{W}_x \leq_m \mathcal{W}_{g(x)} \leq_m A \cup \mathcal{W}_{g(x)}$. But $A \cup \mathcal{W}_{g(x)} \leq_m A$ by hypothesis, so A is m-complete.)

Degtev [1989] has proved that *there are simple, nonhypersimple sets A such that $\boldsymbol{\mathcal{D}_m(\leq a)}$ consists exactly of the m-degrees containing r.e. supersets of A.* In particular, the converse of X.5.14 fails for m-degrees.

However, the converse of X.5.14 almost holds for Turing degrees, in the sense that *if every r.e. superset of A is Turing reducible to it, then A is either hyperhypersimple or T-complete*. This follows from the fact that if A is nonhyperhypersimple, then A has r.e. supersets of any greater r.e. Turing degree (Lachlan [1968d], see the proof of X.6.9.d).

We have seen in Chapter IX that, while the principal filters generated by hyperhypersimple sets are complemented (IX.2.10), those generated by r-maximal sets have no nontrivial complements (IX.2.12). The next result shows that m-degrees reflect a similar contraposition and, together with X.5.14, provides a new proof of IX.2.20.

Proposition X.5.16 (Degtev [1972a]) *If A is r-maximal and B is a nontrivial r.e. superset of it, then the m-degrees of A and B are incomparable.*

Proof. Suppose both $\overline{B}$ and $B - A$ are infinite. To show that A and B are m-incomparable is enough to show that so are $\overline{A}$ and $\overline{B}$. We prove, more generally, that any two subsets B_1 and B_2 of $\overline{A}$ differing infinitely are m-incomparable.

Suppose, for example,

$$y \in B_1 \Leftrightarrow f(y) \in B_2.$$

Consider any $x \in B_2$ and $\{y : f(y) = x\}$. The latter is a recursive subset of $\overline{A}$, and hence it is finite (by simplicity of A). Since B_1 and B_2 differ infinitely, there are infinitely many x such that

1. $x \in \overline{A} \ \wedge \ \{y : f(y) = x\} \subseteq \overline{A} \ \wedge \ (\exists y \neq x)(f(y) = x)$,

because if $f(x) = x$, then x is in either both or none of B_1 and B_2.

Define (inductively on x) a recursive set R by letting $x \in R$ unless one of the following occurs:

- $f(x) \neq x \ \wedge f(x) \in R$
- $f(y) = x \ \wedge \ y < x \ \wedge y \in R$.

The following properties of R are immediate by definition:

2. either x, or $f(x)$, or some y such that $f(y) = x$ is in R
3. if $y \neq x$ and $f(y) = x$, then at most one of x and y is in R.

   Indeed, if $y < x$ and $y \in R$, then $x \notin R$ by the second condition.

   And if $y > x$ and $x \in R$, then $y \notin R$ by the first condition (since $f(y) = x \neq y$ and $f(y) = x \in R$).

From 1 and 3, it follows that $\overline{R} \cap \overline{A}$ is infinite. Then, by r-maximality, $R \cap \overline{A}$ is finite. Consider the recursive set S defined as

$$x \in S \Leftrightarrow f(x) \in R.$$

Then:

- *$S \cap \overline{A}$ is infinite*
  Consider the infinitely many $x \in \overline{A}$ as in 1 above, in particular such that $\{y : f(y) = x\} \subseteq \overline{A}$. Since $R \cap \overline{A}$ is finite, only finitely many such x and y can be in R. By 2 it must then be the case that, for infinitely many x, $f(x) \in R$. Then $x \in S$ by definition, i.e. $S \cap \overline{A}$ is infinite.

- *$\overline{S} \cap \overline{A}$ is infinite*
  Consider the infinitely many $x \in \overline{A}$ as in 1 above, in particular such that, for some $y \in \overline{A}$ different from x, $f(y) = x$. Since $R \cap \overline{A}$ is finite, only finitely many such x can be in R. Thus, for infinitely many x, $f(y) = x \in \overline{R}$. Then $y \in \overline{S}$ by definition, i.e. $\overline{S} \cap \overline{A}$ is infinite.

But this contradicts r-maximality. $\square$

Exercise X.5.17 *If A is a coinfinite r.e. set such that for every nontrivial r.e. superset B of A the m-degrees of A and B are incomparable, then A is r-maximal.* (Hint: if B is recursive and both $B \cap \overline{A}$ and $\overline{B} \cap \overline{A}$ are infinite, $A \cup B$ is a nontrivial r.e. superset of A and $A \cup B \leq_m A$.)

The last connection between structural properties and m-degrees that we investigate is the following.

Proposition X.5.18 (Degtev [1973]) *The m-degree of a dense simple semirecursive set has no minimal predecessor.*

Proof. Let A be a dense simple (III.3.9.b) and semirecursive (III.5.1) set, and let B be a nonrecursive r.e. set such that $B \leq_m A$. We will show that one may suppose that B has the following properties: it is coretraceable (II.6.2.2), and for every recursive function g there is n such that

$$(\forall m > n)(g(b_m) < b_{m+1}),$$

where $\overline{B} = \{b_0 < b_1 < \cdots\}$. Then (II.6.17) there exists a recursive onto function f such that

$$x \in B \Leftrightarrow (\exists y > x)(f(y) \leq f(x)),$$

and $b_n = \max\{x : f(x) = n\}$.

Consider the recursive set

$$x \in R \iff f(x) \text{ even.}$$

Clearly $B \cup R \leq_m B$ and $B \cup \overline{R} \leq_m B$ (for example, the former via

$$t(x) = \begin{cases} b & \text{if } x \in R \\ x & \text{otherwise,} \end{cases}$$

where $b \in B$). Moreover, $B \cup R$ and $B \cup \overline{R}$ are nonrecursive (for example, if $B \cup R$ were recursive then so would be $\overline{B} \cap \overline{R} = \{b_{2n+1} : n \in \omega\}$; but $\overline{B}$ is nonrecursive and retraceable, so it cannot have recursive subsets by II.6.5). To prove that B does not have minimal m-degree, it is enough to show that we cannot have both $B \leq_m B \cup R$ and $B \leq_m B \cup \overline{R}$.

Suppose there are g_1 and g_2 recursive such that

$$x \in B \iff g_1(x) \in B \cup R \qquad \text{and} \qquad x \in B \iff g_2(x) \in B \cup \overline{R}.$$

Let $g(x) = \max\ \{g_1(x), g_2(x)\}$. By the properties of B there is an n such that, for $m > n$, $g(b_m) < b_{m+1}$. We show that B is recursive as follows. Let $b_0, \ldots, b_n$ be given. Suppose that, by induction, we know

$$F = \{y : y < x \wedge y \in \overline{B}\}$$

for $x > b_n$. There are two cases:

- $x \in R$
  We show that
  $$x \in \overline{B} \iff g_1(x) \in F \cap \overline{R}.$$
  If $g_1(x) \in F \cap \overline{R}$, then (since $F \subseteq \overline{B}$) $g_1(x) \in \overline{B} \cap \overline{R}$ and $x \in \overline{B}$.

  If $x \in \overline{B}$, then $x = b_m$ for some $m > n$. Thus $g_1(x) = g_1(b_m) < b_{m+1}$. This means that either $g_1(x) = x$ or $g_1(x) \in F$. Since $x \in \overline{B}$, $g_1(x) \in \overline{R}$ and thus the former cannot happen (because $x \in R$ by case hypothesis). Thus $g_1(x) \in F \cap \overline{R}$.

- $x \in \overline{R}$
  Similarly,
  $$x \in \overline{B} \iff g_2(x) \in F \cap R.$$

This concludes the main proof, but it remains to be shown that B can be supposed to have the stated properties. First consider $C \leq_d B$ simple and with retraceable complement (II.6.16). Then $C \leq_m B$ because B is semirecursive (since A is, and $B \leq_m A$). If we let

$$x \in P \iff (\exists y < x)(h(y) = h(x))$$

where h reduces C to B, then P is recursive and:

- $C \equiv_m C \cup P \leq_1 A$
- $C \cup P$ is dense simple
  If $C \cup P \leq_1 A$ via a one-one recursive function h^* and f dominates $\overline{D}$, then $\max_{x \leq f(n)} h^*(x)$ dominates $\overline{A}$ because h^* is one-one, contradicting the fact that A is dense simple.
- $C \cup P$ is coretraceable
  Note that $\overline{C \cup P} = \overline{C} \cap \overline{P}$ and P is recursive. Then, given φ retracing $\overline{C}$, define ψ as follows: compute $\varphi(x), \varphi^{(2)}(x), \ldots$ and let $\psi(x)$ be the first element of this sequence which is in $\overline{P}$.

Thus we can actually suppose C itself to be dense simple and coretraceable.

It is now enough to show that there is $D \leq_m C$ coretraceable and with the required properties. Since C is dense simple, given f recursive we have $f(n) < c_n$ for almost every n, where $\overline{C} = \{c_0 < c_1 < \cdots\}$. Thus, if we want D such that $f(d_n) < d_{n+1}$ for almost every n, where $\overline{D} = \{d_0 < d_1 < \cdots\}$, it is enough to let

$$\overline{D} = \{c_0 < c_{c_0} < c_{c_{c_0}} < \cdots\},$$

i.e. $d_0 = c_0$ and $d_{n+1} = c_{d_n}$. Clearly D is r.e. and coretraceable, since C is. To show that $D \leq_m C$, let f and g be recursive functions such that

$$c_n = \max\{x : f(x) = n\} \qquad \text{and} \qquad d_n = \max\{x : g(x) = n\}.$$

If we let, for $d \in D$,

$$d(x,0) = \begin{cases} \max\{z : z \leq x \ \wedge \ f(z) = g(x)\} & \text{if } z \text{ exists} \\ d & \text{otherwise} \end{cases}$$

then:

- if $x = d_n$, then $x > c_n$ and so $d(d_n, 0) = c_n$
- if $g(x) = 0$, then $x \leq c_0 = d_0$ and so $d(x,0) = x$ or $d(x,0) = d$.

In particular, $d(d_n, 0) = c_n$ and

$$\text{if } g(x) = 0, \text{ then } [x \in C \ \Leftrightarrow \ d(x,0) \in D].$$

We can then use $d(x,0)$ to iterate, and let

$$d(x,1) = \begin{cases} \max\{z : d(x,0) < z \leq x \ \wedge \ f(z) = d(x,0)\} & \text{if } z \text{ exists} \\ d & \text{otherwise.} \end{cases}$$

Then, similarly:

- if $x = d_n$ $(n \geq 1)$, then $d(x,0) = c_n$ and so $d(x,1) = c_{c_n}$
- if $g(x) = 1$, then $x \in C$ if and only if $d(x,1) \in D$.

Thus, in general, we let

$$d(x, n+1) = \begin{cases} \max\{z : d(x,n) < z \leq x \ \wedge \ f(z) = d(x,n)\} & \text{if } z \text{ exists} \\ d & \text{otherwise.} \end{cases}$$

Then $x \in C$ if and only if $d(x, g(x)) \in D$. $\square$

Corollary X.5.19 (Ershov [1969], Lachlan [1972a]) *There is an r.e. m-degree without minimal predecessors.*

Proof. This follows from X.5.18, by the existence of a dense simple semirecursive set (a direct construction is given on p. 395, while a natural example is exhibited in IX.2.34). $\square$

Originally, Ershov [1969] proved the corollary by modifying Lachlan's construction of a hyperhypersimple set without maximal supersets (IX.2.28.f), building on X.5.14. Lachlan [1972a] instead embedded $1 + \omega^*$ (i.e. the reverse ordering of the natural numbers with a least element added to it) as an initial segment. This can also be obtained from the proof of X.5.25 (see the comments following it).

Minimal degrees

We have seen in X.5.12 that one can build minimal r.e. m-degrees by constructing η-maximal sets. The following exercises exploit this possibility, and provide a good deal of minimal r.e. m-degrees.

Exercises X.5.20 a) *Every nonzero r.e. T-degree contains a minimal r.e. m-degree.* (Lachlan [1972]) (Hint: given an r.e. set B, we build an η-maximal set A such that $B \equiv_T A$. One might think of modifying the construction of III.5.19, forgetting about the work done for semirecursiveness, by using permitting to obtain $A \leq_T B$, and coding for $B \leq_T A$. At stage $s+1$, having boxes A_n^s, we might put A_n^s in A when n shows up in B. The problem is that this does not allow us to recover B recursively in A, as in usual permitting, because we do not know what the final A_n is going to be. We thus modify this approach a little, deciding to put in A, for the sake of coding and when n shows up in B, a number between $f(n)$ and $f(n+1)$, for a given recursive function f.

The construction can be pictured as follows. We start with boxes $A_x^0 = \{x\}$. The boxes $\{A_x^s : f(n) \leq x < f(n+1)\}$ form the n-th block. During the construction the elements of a box can either go into A for the coding, or into a box with lower index (actually, a box of a preceding block) for η-maximality.

To make A η-maximal, when we see a nonempty box A_x^s which does not intersect some $\mathcal{W}_{e,s}$ for $e \leq x$, but such that some A_y^s in a following block does, then we send the elements of A_y^s into A_x, and A_y will remain empty from now on.

To code B into A, when $n \in B_{s+1} - B_s$ we take the smallest nonempty box in the n-th block and put it into A. To know if $n \in B$, note that if A_x^s is nonempty then x is in it. For all x such that $f(n) \leq x < f(n+1)$, see if $x \in A$. If so, look for the stage $s_x + 1$ at which it goes into A. Then

$$n \in B \ \Leftrightarrow\ x \in A_x^{s_x} \text{ for some such } x.$$

Note that the condition $x \in A_x^{s_x}$ tells us that x goes into A for the coding of n, and not because it is in the wrong box (in which case it could go into A for the coding of another element).

We have to choose f in such a way that in the n-th block there is always a nonempty box, so that the coding is always possible. Suppose we already have $f(n)$. For each $x < f(n)$ we might want to increase the x- th state of A_x, and to do this we might need one box for each $e \leq x$. It is possible that we take all these boxes from the n-th block. Moreover, we need one more box to put into A for the coding, and another one to ensure that $\overline{A}$ is η-infinite. We may thus define $f(0) = 0$ and $f(n+1) = f(n) + \sum_{x<f(n)}(x+1) + 2$.

We still have to argue that $A \leq_T B$. Let $p(n)$ be the index of the block that contains A_n. By construction $x \in A_x^0$, and at later stages either $x \in A$ or $x \in A_i^s$ for $i \leq x$. But A_i^s goes into A at stage $s+1$ only if $p(i) \in B_{s+1} - B_s$. If s is a stage at which all elements $p(0), \ldots, p(x)$ that are in B are already in B_s, then $x \in A$ if and only if $x \in A_s$.)

b) *Every nonzero r.e. T-degree contains infinitely many minimal r.e. m-degrees.* (Kobzev [1973a]) (Hint: modify part a) by using priority and permitting.)

While the use of η-maximality for the construction of minimal r.e. m-degrees is elegant, one sometimes needs more flexibility. We now give *a direct construction of a minimal r.e. m-degree*, using a framework similar to that for η-maximality. In particular, we will obtain $\overline{A}$ as a sequence of boxes A_n, each of them being the limit of recursive approximations A_n^s.

We want

$$B \leq_m A \ \wedge\ B \text{ nonrecursive } \Rightarrow\ A \leq_m B.$$

Suppose

$$z \in B \ \Leftrightarrow\ f(z) \in A.$$

The idea is to ensure that, for almost every x, either $x \in A$ or, for some z, s and n,

$$f(z) \in A_n^s \quad \wedge \quad x \in A_n^s, \tag{X.1}$$

i.e. some $f(z)$ goes in the same box as x.

If we choose $b \in B$ and let

$$g(x) = \begin{cases} b & \text{if } x \in A \text{ first} \\ z & \text{if } f(z) \in A_n^s \wedge x \in A_n^s \text{ first,} \end{cases}$$

then

$$x \in A \Leftrightarrow g(x) \in B$$

because, since by construction we put boxes into A,

$$g(x) = z \in B \Leftrightarrow f(z) \in A \Leftrightarrow A_n^s \subseteq A \Leftrightarrow x \in A.$$

To achieve X.1 we simply have to ensure that, given e, if (range $\varphi_e) \cap \overline{A}$ is infinite then each box from a certain point on contains an element of the range of φ_e (note that if B is nonrecursive then, since

$$z \in \overline{B} \Leftrightarrow f(z) \in \overline{A},$$

(range $f) \cap \overline{A}$ is infinite).

Since the range of φ_e is an r.e. set, in the proofs below we simply make sure that whenever $\mathcal{W}_i \cap \overline{B}$ is infinite, $\mathcal{W}_i$ intersects every box from a certain point on. This is how η-maximality comes into the picture.

The next result provides half of the ideas needed to build finite initial segments, and shows that *there is no r.e. m-degree* $\boldsymbol{a} < \mathbf{0}'_{\boldsymbol{m}}$ *bounding all minimal r.e. m-degrees.*

Proposition X.5.21 (Ershov and Lavrov [1973]) *For every r.e. m-degree* $\boldsymbol{c}$ *such that* $\mathbf{0}_{\boldsymbol{m}} < \boldsymbol{c} < \mathbf{0}'_{\boldsymbol{m}}$*, there is a minimal r.e. m-degree* $\boldsymbol{a}$ *incomparable with it.*

Proof. Given an r.e. set C nonrecursive and such that $\mathcal{K} \not\leq_m C$, we want to build an r.e. set A of minimal m-degree such that $A \not\leq_m C$. This is enough because, by minimality, if $C \leq_m A$, then $C \equiv_m A$. From $A \not\leq_m C$ it thus follows that $C \not\leq_m A$, too. Note also that the condition $A \not\leq_m C$ automatically implies that A is not recursive.

To obtain $A \not\leq_m C$, we use the coding method (X.5.1) and code $\mathcal{K}$ into A as long as it appears that $A \leq_m C$. If the coding is recoverable by m-reductions, then

$$A \leq_m C \Rightarrow \mathcal{K} \leq_m A \leq_m C,$$

contradicting the hypothesis on C.

If we use the technique sketched in X.5.20, we cannot achieve the coding. The problem is that if $x_0^e < x_1^e < \cdots$ are the elements designed to have

$$n \in \mathcal{K} \Leftrightarrow x_n^e \in A$$

when $A \leq_m C$ via φ_e, the strategies for coding and η-maximality might conflict in the following way. We may want to put x_n^e and x_m^e in the same box for η-maximality, when only one of n and m is in $\mathcal{K}$, but this would ruin the coding, since then either both or none of x_n^e and x_m^e will be in A. We thus have to be a little more sophisticated.

We start with $S = \omega$, and decide to devote e.g.

$$Q = \{2x : x \in \omega\}$$

to the coding for the requirement

$$P_0 \ : \ A \not\leq_m C \text{ via } \varphi_0.$$

Thus we put $2n$ into A at stage $s+1$ if we are considering P_0 (as we will infinitely often), $n \in \mathcal{K}_s$ and $n < l_p(0,s)$, where (as in X.5.1),

$$l_p(e,s) = \max\{z : (\forall y < z)[y \in A_s \ \Leftrightarrow \ \varphi_{e,s}(y) \in C_s]\}.$$

Since the coding succeeds for the usual reasons (i.e. because $\mathcal{K} \not\leq_m C$), only a finite subset F of Q will ever go into A.

We now work on

$$R = S - Q = \{2x+1 : x \in \omega\}$$

for the remaining requirements. We consider the 0-th requirement for minimality, and we thus look at $\mathcal{W}_0$. Suppose

$$x \in B \ \Leftrightarrow \ f(x) \in A,$$

where f is a recursive function with range $\mathcal{W}_0$.

If $\mathcal{W}_0 \cap R$ is finite, then B will be recursive independently of how we define A (since $\mathcal{W}_0 \cap A$ is finite, because $A \cap Q$ and $\mathcal{W}_0 \cap R$ are both finite).

If $\mathcal{W}_0 \cap R$ is infinite, we want to ensure that $A \leq_m B$ and also leave enough space to satisfy the other requirements. This can easily be achieved, as follows. Choose an infinite recursive set $S' \subseteq \mathcal{W}_0 \cap R$ to work on in the future, and put all the elements of $R - S'$ into A. Then $A \leq_m B$ via the following g, where $b_0 \in B$ and $b_1 \in \overline{B}$:

$$g(x) = \begin{cases} \mu z.\,(f(z) = x) & \text{if } x \in S' \\ b_0 & \text{if } x \in R - S' \text{ or } x \in F \\ b_1 & \text{if } x \in Q - F \end{cases}$$

(recall that $F = A \cap Q$). This works because $S' \subseteq \mathcal{W}_0$. More precisely, if $x \in S'$, then $g(x)$ is defined and

$$x \in A \ \Leftrightarrow \ f(z) \in A \ \Leftrightarrow \ z \in B \ \Leftrightarrow \ g(x) \in B.$$

The problem with this construction is that the split into the two cases ($\mathcal{W}_0 \cap R$ infinite or not) is not effective, while we want A to be r.e. We thus approximate the construction by playing the two strategies simultaneously, to allow for both possibilities (this is reminiscent of the e-state method). We divide R into two sets S_0 and S_1, and put into S_0 elements of $\mathcal{W}_0 \cap R$ in increasing order, so that S_0 will turn out to be either finite or (infinite but) recursive. In the first case, we will continue to work on S_1. In the second case, we will work on S_0, but then we will have to put elements of S_1 into A (since now S' and $R - S'$ above are replaced by S_0 and S_1, respectively).

Thus at stage $s+1$, when considering the 0-th minimality requirement (as we will infinitely often), we put:

- into $S_{0,s+1}$ the smallest element of $S_{1,s+1}$ which is in $\mathcal{W}_{0,s} - A_s$ and greater than max $S_{0,s}$ (to have S_0 recursive)
- into A the elements of $S_{1,s+1}$ smaller than it.

If S_0 turns out to be infinite, we will have $S_0 \subseteq \mathcal{W}_0$ and $S_1 \subseteq A$. Otherwise, only a finite number of elements of S_1 go into A for the second clause above.

We can now play the same strategy inside S_0 and S_1, and so on. In general, we will have a tree of recursive boxes S_σ (σ a string of 0's and 1's). If

$$S_\sigma = \{x_0^\sigma < x_1^\sigma < \cdots\}$$

we can devote

$$Q_\sigma = \{x_0^\sigma < x_2^\sigma < \cdots\}$$

to the coding for the requirement P_e with $e = |\sigma|$, and define $S_{\sigma*0}$ and $S_{\sigma*1}$ inside

$$R_\sigma = \{x_1^\sigma < x_3^\sigma < \cdots\}.$$

The remarks given above are easily modified to show that the requirements for minimality and for the coding are satisfied. $\Box$

Corollary X.5.22 Capping Theorem for $\mathcal{R}_m$. *Every nontrivial r.e. m-degree is part of a minimal pair.*

Initial segments

The next result provides the second half of the ideas required to build finite initial segments of the r.e. m-degrees, and it is an effective analogue of VI.3.1.

Proposition X.5.23 (Lachlan [1972]) *Every r.e. m-degree $\mathbf{c} < \mathbf{0}'_m$ has a strong minimal cover in the r.e. m-degrees.*

Proof. The construction is a modification of that in X.5.21. Given an r.e. set C such that $\mathcal{K} \not\leq_m C$, we want to build an r.e. set A such that:

1. $C \leq_m A$
2. $A \not\leq_m C$
3. $B \leq_m A \;\Rightarrow\; B \leq_m C$ or $A \leq_m B$.

We easily obtain 1 by coding C into, say, $Q_\emptyset$. We might then devote Q_σ to the coding for the requirement

$$P_e \;:\; A \not\leq_m C \text{ via } \varphi_e$$

with $e = |\sigma| - 1$, and obtain 2 as before.

However, if we then proceed as in the construction of X.5.21, we do not obtain A as a *strong* minimal cover of C. Suppose, for example,

$$x \in B \;\Leftrightarrow\; f(x) \in A,$$

where f is a recursive function with range $\mathcal{W}_0$. We have two cases:

1. *S_0 is finite*
   Then we easily obtain $B \leq_m C$, as follows. Given x, consider $f(x)$ (which is in $\mathcal{W}_0$).

   If $f(x) \in Q_\sigma$, then $f(x) = x^\sigma_{2n}$ for some n, and

   $$x \in B \;\Leftrightarrow f(x) \in A \;\Leftrightarrow n \in C,$$

   by the coding strategy.

   If $f(x) \in R_\sigma$, then the only interesting case is when $f(x) \in S_1$ (since S_0 is finite by case hypothesis). But then $f(x) \in A$, and so $x \in B$ (because by construction every element of S_1 which is in $\mathcal{W}_0 - A$ goes into S_0).

   Thus $B \leq_m C$, via the function that associates n with x in the first case, and a fixed element $c \in C$ in the second.

2. *S_0 is infinite*
   We now want $A \leq_m B$. The reduction is easily obtained when $x \in S_0$ since (as $S_0 \subseteq \mathcal{W}_0$) it is enough to consider $g(x) = \mu z.\,(f(z) = x)$ to obtain

   $$x \in A \;\Leftrightarrow\; g(x) \in B.$$

   If $x \in S_1$, then we have $S_1 \subseteq A$ by construction and so, again, there is no problem.

If $x \in Q_\sigma$, then $x = x_{2n}^\sigma$, but we only know

$$x \in A \Leftrightarrow n \in C.$$

This gives $A \leq_m C$, but there is no reason to suppose in general that $C \leq_m B$ (this would be enough if one just wanted a minimal cover, not a strong one).

We can avoid this last difficulty by modifying the construction in such a way as to have $C \leq_m S_0$, since then a procedure like the one just sketched will suffice. This is easily obtained if we decide to code C not only in $Q_\emptyset$, but also in Q_0 and, in general, in every Q_σ. However, since we also have the codings for P_e, we simply decide to devote Q_σ to both codings, as follows. If

$$Q_\sigma = \{x_0^\sigma < x_2^\sigma < \cdots\},$$

then we code C on $\{x_{4z}^\sigma\}_{z\in\omega}$, i.e.

$$x_{4z}^\sigma \in A \Leftrightarrow z \in C,$$

and we use $\{x_{4z+2}^\sigma\}_{z\in\omega}$ for the coding relative to P_e, when $e = |\sigma|$. As usual, this last part contributes only finitely many elements to A. $\Box$

By combining the proofs of X.5.21 and X.5.23, one immediately obtains the following result.

Proposition X.5.24 (Ershov and Lavrov [1973]) *For any pair $\boldsymbol{b}$ and $\boldsymbol{c}$ of r.e. m-degrees such that $\boldsymbol{c} < \boldsymbol{0'_m}$ and $\boldsymbol{c} \not\leq \boldsymbol{b}$, there is a strong minimal cover of $\boldsymbol{b}$ in the r.e. m-degrees incomparable with $\boldsymbol{c}$.*

This result says that one can always find (except in trivial situations) minimal covers of given m-degrees incomparable with any other given m-degree, and it is sufficient to prove the following analogue of VI.2.3.

Theorem X.5.25 (Lachlan [1972a]) *The topped finite initial segments of the r.e. m-degrees are exactly the finite distributive lattices.*

Proof. The condition is necessary by VI.1.10. Conversely, given a finite distributive lattice L (with ordering $\sqsubseteq$), we want to define an ideal $\mathcal{L}$ of the r.e. m-degrees isomorphic to L. Let 0 be the smallest element of L, and define $I_0 = \{0\}$, $\mathcal{I}_0 = \{\boldsymbol{0_m}\}$ and $\varphi_0(0) = \boldsymbol{0}_m$.

For any given n, let I_n and $\mathcal{I}_n$ be finite ideals of L and of the r.e. m-degrees isomorphic via φ_n. If $L - I_n \neq \emptyset$, take $a \in L - I_n$ such that a is a strong minimal cover of b, for some $b \in I_n$. Such an a exists because L is finite and I_n is an ideal of it. More precisely, take any $c \in L - I_n$. If the elements strictly below c are all in I, they must have a l.u.b. in I (because I is an ideal) and c is a strong minimal cover of it. Otherwise, there is a $c_1 \sqsubset c$ such that $c_1 \in L - I_n$, so one can start again with c_1 in place of c and so on. The process must eventually end, because L is finite.

We extend φ_n to φ defined on $I_n \cup \{a\}$ as follows (where d is the l.u.b. of I_n in L):

- if $b = d$, then let $\varphi(a)$ be a strong minimal cover of $\varphi_n(b) = \varphi_n(d)$ (which exists by X.5.23, because $\mathcal{I}_n$ is a finite ideal and thus its l.u.b. cannot be $\mathbf{0}'_m$ by X.5.8)
- if $b \sqsubset d$, then let $\varphi(a)$ be a strong minimal cover of $\varphi_n(b)$ incomparable with $\varphi_n(d)$ (which exists by X.5.24).

Now let I_{n+1} and $\mathcal{I}_{n+1}$ be the ideals of L and of the r.e. m-degrees generated, respectively, by $I_n \cup \{a\}$ and $\mathcal{I}_n \cup \{\varphi(a)\}$. Then the proof of VI.3.6 shows that φ can be extended to an isomorphism φ_{n+1} of I_{n+1} and $\mathcal{I}_{n+1}$. Since L is finite, the process ends in finitely many steps. $\Box$

The proof given above is due to Degtev [1979], and it actually shows that any finite initial segment $\mathcal{I}$ of the r.e. m-degrees can be extended to an initial segment $\mathcal{L}$ isomorphic to L, for any finite distributive lattice L with an ideal I isomorphic to $\mathcal{I}$ (Denisov [1978]). As a consequence, we obtain as initial segments of the r.e. m-degrees, all countable uppersemilattices which can be obtained as unions of chains of finite distributive lattices, each one being an ideal of the following (Lachlan [1972a]).

A countable lattice $(L, \sqsubseteq, \sqcap, \sqcup)$ is 'effective' if there is an enumeration (not necessarily effective) $\{a_x\}_{x\in\omega}$ of it such that $a_x \sqsubseteq a_y$ is a Π^0_2 relation of x and y, and for some recursive functions f and g,

$$a_x \sqcup a_y = a_{f(x,y)} \qquad \text{and} \qquad a_x \sqcap a_y = a_{g(x,y)}.$$

A distributive uppersemilattice is 'effective' if it is the limit of an r.e. sequence of finite, uniformly effective distributive lattices. Lachlan [1972a] has given a constructive version of VI.2.5, by showing that *the topped initial segment of the r.e. m-degrees are exactly the effective distributive uppersemilattices*. In particular, one obtains the following constructive version of VI.3.3: *the order-types of the linearly ordered initial segments of the r.e. m-degrees are exactly the recursive linear orderings with least element.*

Exercise X.5.26 *The ordinals of the well-ordered initial segments of the r.e. m-degrees are exactly the ordinals $\leq \omega_1^{ck}$.* (Lachlan [1972a]) (Hint: the necessity of the condition follows from the fact that if $\boldsymbol{a}$ is r.e. and the m-degrees below it are isomorphic to an ordinal, then this is an arithmetical ordinal and hence, by a theorem of Spector proved in Volume III, a recursive ordinal. Conversely, an initial segment isomorphic to ω_1^{ck} can be built by starting with $\mathbf{0}_m$, and iterating strong minimal covers. That the latter exist at recursive limit ordinals is obtained in a way similar to X.5.23, using the fact that there is a recursive enumeration of representatives of the m-degrees in the chain.)

Ershov and Lavrov [1973] have proved that *an ideal of $\boldsymbol{\mathcal{R}_m}$ has a strong minimal cover if and only if it is bounded below $\mathbf{0}'_m$ and is Σ^0_3*. The conditions are obviously necessary, since $\mathbf{0}'_m$ is not a minimal cover (X.5.2) and for every r.e. set A the index set $\{x : \mathcal{W}_x <_m A\}$ is Σ^0_3 (X.9.14).

Global properties

VI.4.1 admits the following constructive analogue.

- **Characterization of $\boldsymbol{\mathcal{R}_m}$ (Denisov [1978])** *Up to isomorphism, $\boldsymbol{\mathcal{R}_m}$ is the only structure with the following properties:*

  1. *it is an effective distributive uppersemilattice with least and greatest element*
  2. *every principal ideal $\mathcal{I}$ not containing the greatest element can be extended to an ideal $\mathcal{L}$ isomorphic to L, for any effective distributive uppersemilattice L with an ideal effectively isomorphic to $\mathcal{I}$.*

As in the case of VI.4.1, this result contains all the information about the structure of r.e. m-degrees, and from it one can derive solutions to a number of algebraic problems. In particular, one obtains the following analogues of VI.4.2–6.

- **Strong homogeneity.** *For any r.e. m-degree $\boldsymbol{a} < \mathbf{0}'_{\boldsymbol{m}}$, the cone of r.e. m-degrees above $\boldsymbol{a}$ is isomorphic to the r.e. m-degrees.*

- $\mathbf{0}_m$ *and* $\mathbf{0}'_m$ *are the only r.e. m-degrees fixed under every automorphism, as well as the only definable r.e. m-degrees.*

- *Every definable nontrivial set of r.e. m-degrees is infinite.*

- *There are $2^{\aleph_0}$ automorphisms of $\boldsymbol{\mathcal{R}_m}$.*

The only problem left open by the previous results is the characterization of the complexity of the theory of $\boldsymbol{\mathcal{R}_m}$, which is solved by the following analogue of VI.4.8.

Theorem X.5.27 Nies' Theorem (Nies [1994]) *The first-order theory of $\mathcal{R}_m$ has the same degree (and actually the same isomorphism type) as the theory of First-Order Arithmetic.*

Proof. We prove that the two-theories have the same m-degree by interpreting each in the other, thus providing faithful translations that will preserve theorems. Since the translations will actually be one-one, the theories will have the same 1-degree, and hence will be recursively isomorphic by III.7.13.

One direction is clear, since every formula about the ordering of the r.e. m-degrees can be interpreted in a natural way as a formula about r.e. sets of integers, and hence about their r.e. indices. Thus the theory of r.e. m-degrees is interpretable in First-Order Arithmetic.

For the converse, we want to show that First-Order Arithmetic is interpretable in $\mathcal{R}_m$. The proof elaborates the ideas used in VI.4.8, and is devoted to showing on the one hand that we can express in $\mathcal{R}_m$ the fact that given parameters code a standard model of arithmetic, and on the other hand that such parameters exist in the r.e. m-degrees.

Having done this, then a sentence φ of First-Order Arithmetic is true if and only if the sentence φ^* of $\mathcal{R}_m$ is true, where φ^* states that for any parameters coding a standard model of arithmetic in the r.e. m-degrees, the translation of φ holds. Since φ^* can be effectively obtained from φ, we thus have an m-reduction of First-Order Arithmetic to the theory of $\mathcal{R}_m$.

It remains to deal with models of arithmetic:

1. *standard models*

   A *model of arithmetic* is a structure $M = \langle A, R, f_1, f_2, f_3, a\rangle$ such that:

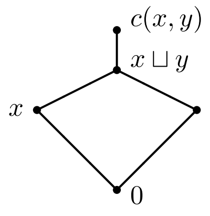
   (a) A is a countable set

   (b) R is a total ordering on A with a first element a and a one-one successor given by f_1, and such that a is not the successor of any element of A, and every element different from a has an immediate predecessor

   (c) f_2 and f_3 satisfy the usual recursive definitions of sum and product.

   A *standard model of arithmetic* is a model in which R is a well-ordering.

   To code a standard model we use the methods of VI.4.7, to whose proof we refer, to code relations by graphs, and graphs by ideals of distributive lattices. In particular, given a graph on a countable domain, the atoms of the lattice correspond to the elements of the domain, and the pair of atoms $\{x, y\}$ is coded by an element $c(x, y)$ added above $x \sqcup y$, as in the following picture.



In the present case the ideals will just be initial segments of $\boldsymbol{\mathcal{R}_m}$, which are automatically distributive, and the atoms of the lattices will thus be minimal r.e. m-degrees.

A standard model of m-incomplete r.e. m-degrees (with universe consisting of minimal r.e. m-degrees) can be embedded in $\boldsymbol{\mathcal{R}_m}$, by the methods of X.5.25: indeed, despite the fact that the lattices needed for the coding are infinite ones, the picture above shows that they are locally finite, and can thus be uniformly built by successive finite extensions. Moreover, the uniformity of the construction shows that one thus obtains recursively presented ideals of r.e. m-degrees, that admit exact pairs by the methods of X.5.5. Thus there are parameters $\vec{\boldsymbol{a}}$ coding A, and $\vec{\boldsymbol{c}}$ coding R and the graphs of the functions f_1, f_2, and f_3.

Moreover, given degrees $\vec{\boldsymbol{a}}$ coding a countable set of minimal r.e. m-degrees (intended to be the universe A of a standard model M of arithmetic in $\boldsymbol{\mathcal{R}_m}$), we can say in a first-order statement of $\boldsymbol{\mathcal{R}_m}$ that given degrees $\vec{\boldsymbol{c}}$ code a relation and (the graphs of) three functions on the set coded by $\vec{\boldsymbol{a}}$, satisfying the requirements for a model of arithmetic. This is because the definition of a model requires only finitely many first-order properties.

The whole problem is thus to express the fact that the model is standard, since the notion of well-ordering is a second-order one. This would apparently require quantification over all subsets of A, and hence over the parameters coding them. But there are uncountably many such subsets, and thus not all can be coded by r.e. parameters. To be able to handle standard models we then need to proceed in a more indirect way.

The idea is, given any two models of arithmetic M_1 and M_2 coded by parameters $\vec{\boldsymbol{a}}_1$, $\vec{\boldsymbol{c}}_1$ and $\vec{\boldsymbol{a}}_2$, $\vec{\boldsymbol{c}}_2$, to define in a uniform way, with additional parameters $\vec{\boldsymbol{d}}$, a partial map $f : A_1 \to A_2$ that sends the interpretation $\boldsymbol{n}_1$ of n in M_1 to the interpretation $\boldsymbol{n}_2$ of n in M_2 (in other words, the restriction of f to the standard part of M_1 provides an isomorphism to the standard part of M_2).

Then one can say that M_1 coded by $\vec{\boldsymbol{a}}_1$, $\vec{\boldsymbol{c}}_1$ is standard if and only if, for any parameters $\vec{\boldsymbol{a}}_2$, $\vec{\boldsymbol{c}}_2$ coding a model M_2, there are parameters $\vec{\boldsymbol{d}}$ coding a map as above that is total on A_1, and order preserving.

Indeed, if M_1 is standard then, for any M_2, the partial map f defined above is total (because A_1 has only standard elements, and f is defined on all of them), and order preserving (because f is then an isomorphism of all of A_1 and the standard part of A_2).

Conversely, if such a map is always total and order preserving, it is so in particular for any standard model M_2: then M_1 must be standard too, because all the elements of A_1 are mapped in an order preserving way into the elements of A_2, which is standard.

2. *isomorphisms between standard parts of models*
To define a map $f : A_1 \to A_2$ providing an isomorphism of the full standard parts of M_1 and M_2, it is enough to uniformly define the finite approximations σ providing isomorphisms of finite initial segments of the standard parts. Then one can define the full (partial) f by saying that $f(\boldsymbol{i}) = \boldsymbol{j}$ if and only if there are degrees $\boldsymbol{x}$ in A_1 and $\boldsymbol{y}$ in A_2, such that $\boldsymbol{i}$ belongs to the initial segment $\mathbf{0}_1, \ldots, \boldsymbol{x}$ of A_1, $\boldsymbol{j}$ belongs to the initial segment $\mathbf{0}_2, \ldots, \boldsymbol{y}$ of A_2, and there is an isomorphism σ of these initial segments, such that $\sigma(\boldsymbol{i}) = \boldsymbol{j}$. Then f is the union of a first order definable collection of isomorphisms of initial segments.

To define an isomorphism σ of the initial segments

$$\mathbf{0}_1, \ldots, \boldsymbol{x} \qquad \text{and} \qquad \mathbf{0}_2, \ldots, \boldsymbol{y}$$

of A_1 and A_2, we would like to construct an auxiliary standard model M_3 with

$$\mathbf{0}_1, \ldots, \boldsymbol{x}, \mathbf{0}_2, \ldots, \boldsymbol{y}$$

as an initial segment, but this is impossible in general: indeed, M_1 and M_2 are arbitrary models, and it may be that the two initial segments of A_1 and A_2 overlap, in the sense that the same r.e. m-degree can be the interpretation of an integer (not necessarily the same one) in both models.

However, since the initial segments are finite, there exists a degree $\boldsymbol{k}$ in A_2 such that all $\boldsymbol{i}$ smaller than or equal to $\boldsymbol{x}$ in A_1 are strictly smaller than $\boldsymbol{k}$ in A_2, and the least such degree in A_1 is definable from the parameters needed to define M_1 and M_2, plus $\boldsymbol{x}$ and $\boldsymbol{y}$. We would thus like to construct an auxiliary standard model M_3 whose first integers are the degrees

$$\mathbf{0}_1, \ldots, \boldsymbol{x}, \boldsymbol{k}, \ldots, \boldsymbol{k} +_2 \boldsymbol{y},$$

where $+_2$ is the interpretation of the sum in M_2, and define σ for $\boldsymbol{i} \leq_1 \boldsymbol{x}$ and $\boldsymbol{j} \leq_2 \boldsymbol{y}$ as

$$\sigma(\boldsymbol{i}) = \boldsymbol{j} \quad \Leftrightarrow \quad \boldsymbol{k} +_2 \boldsymbol{j} = (\boldsymbol{x} +_3 \mathbf{1}_3 +_3 \boldsymbol{i}),$$

where $\leq_1$ and $\leq_2$ are the interpretations of the order of integers in M_1 and M_2, and $+_3$ is the interpretation of the sum of integers in M_3. This definition of σ uses $\boldsymbol{x}$, $\boldsymbol{y}$ and the parameters needed to define M_3.

3. *existence of the auxiliary models*
   We still have to show the existence of an M_3 as above, given $\boldsymbol{x}$ and $\boldsymbol{y}$. The construction of standard models of arithmetic given in the first part of the proof proceeds by successive finite extensions: it is thus possible to build a standard model of arithmetic in the same way, starting from finitely many given r.e. minimal m-degrees as interpretations of the first natural numbers. In other words, we can fix ahead of time a finite initial segment of the model, and extend it to a full standard model.

   In the application needed above, the given initial segment is obtained from initial segments of M_1 and M_2, and can thus be defined from the parameters defining the two models. $\square$

The next result was originally obtained from X.5.25.

Corollary X.5.28 (Lachlan [1972a]) *The first-order theory of $\boldsymbol{\mathcal{R}_m}$ is undecidable and not axiomatizable.*

Turning to subclasses of sentences, one has that *the two-quantifier theory of $\boldsymbol{\mathcal{R}_m}$ is decidable* (Degtev [1979]), while *the three-quantifier theory of $\boldsymbol{\mathcal{R}_m}$ is undecidable* (Nies [1996]).

One can also look at the structures $\boldsymbol{\mathcal{D}_m^a}$ and $\boldsymbol{\mathcal{R}_m^a}$ defined as $\boldsymbol{\mathcal{D}_m}$ and $\boldsymbol{\mathcal{R}_m}$, but with many-one reductions recursive in the Turing degree $\boldsymbol{a}$. Nies [1996a] has proved that, although nothing happens globally because *$\boldsymbol{\mathcal{D}_m^a}$ and $\boldsymbol{\mathcal{D}_m}$ are isomorphic for any $\boldsymbol{a}$*, locally one has the following versions of V.7.13 and V.7.16:

- *if $\boldsymbol{\mathcal{R}_m^a}$ is elementarily equivalent to $\boldsymbol{\mathcal{R}_m}$, then $\boldsymbol{a}$ is low$_3$*
- *if $\boldsymbol{\mathcal{R}_m^a}$ is isomorphic to $\boldsymbol{\mathcal{R}_m^b}$, then $\boldsymbol{a}$ and $\boldsymbol{b}$ have the same triple-jump.*

X.6 Turing Degrees

In this section we deal with r.e. ***wtt*-degrees** and ***T*-degrees**, and comment on their similarities and differences.

A thorough knowledge of Section 3, to which we refer for notation, is required for some of the proofs.

Incomparable degrees

We already know that there are incomparable r.e. Turing degrees (X.1.7). We now show how to extend the coding method (introduced in a weak version in the proof of X.5.1, to which we refer below) to produce degrees incomparable with a given nontrivial one.

Theorem X.6.1 (Sacks [1963a], Yates [1966a]) *For every r.e. degree* $\boldsymbol{c}$ *such that* $\mathbf{0} < \boldsymbol{c} < \mathbf{0}'$ *there is an incomparable r.e. degree* $\boldsymbol{a}$.

Proof. Let C be an r.e. nonrecursive set such that $\mathcal{K} \not\leq_T C$. We want to build an r.e. set A such that

$$\begin{array}{rcl} P_e & : & A \not\simeq \{e\}^C \\ N_e & : & C \not\simeq \{e\}^A. \end{array}$$

To satisfy N_e we use the Sacks agreement method. In particular, we define a length of agreement function l_n for negative requirements.

To satisfy P_e we would like (as in X.5.1) to code $\mathcal{K}$ into A as long as it seems that $A \simeq \{e\}^C$, in particular by defining a length of agreement function l_p for positive requirements:

$$l_p(e,s) \;=\; \max\{z : (\forall y < z)[A_s(y) \simeq \{e\}_s^{C_s}(y)]\}.$$

The problem now is that this might produce infinitely many injuries, since we might have $\lim_{s\to\infty} l_p(e,s) = \infty$ even if $A \not\simeq \{e\}^C$. For example, if z_e is the only element on which they disagree, $\{e\}^C$ might diverge on z_e while, for infinitely many s, $A_s(z_e) \simeq \{e\}_s^{C_s}(z_e)$, so that $\lim_{s\to\infty} l_p(e,s) = \infty$.

A use of the modified restraint function $\hat{r}$ of X.3.4 in the construction is thus forced upon us, but it is not enough. If we use the same coding as in X.5.1 then, when $\lim_{s\to\infty} l_p(e,s) = \infty$, we would actually code $\mathcal{K}$ into $A^{[e]}$, and thus A would be T-complete. We thus have to concoct a more sophisticated coding that contributes enough elements to satisfy P_e, without however endangering N_e (note that we want to satisfy N_e by the Injury Lemma X.3.2, and for this we want $C \not\leq_T \hat{I}_e$; since $\hat{I}_e \leq_T A^{[<e]}$, we need to control the coding).

If we use $\{\hat{e}\}^C$ in place of $\{e\}^C$ (X.3.4) and define $\hat{l}_p$ accordingly, i.e.

$$\hat{l}_p(e,s) \;=\; \max\{z : (\forall y < z)[A_s(y) \simeq \{\hat{e}\}_s^{C_s}(y)]\},$$

at least we will have

$$A \not\simeq \{e\}^C \;\Rightarrow\; \liminf_{s\to\infty} \hat{l}_p(e,s) < \infty.$$

The additional idea is to consider $\hat{l}_p(e,s)$ dynamically and to stop putting elements into $A^{[e]}$ when it appears that $\hat{l}_p(e,s)$ has dropped back. Thus, instead of simply considering $\langle e, x\rangle$ we actually consider $\langle e, x, s\rangle$ (where, for a fixed e, $\langle e, x, s\rangle$ unfolds the e-th column).

The construction is as follows. We start with $A_0 = \emptyset$. At stage $s+1$, we put $\langle e, x, t\,\rangle$ in A if

$$x \in \mathcal{K}_s \ \wedge\ (\forall i \leq e)[\langle e, x, t\,\rangle > \hat{r}(i, s)] \ \wedge\ (\forall u)[t \leq u \leq s \ \Rightarrow\ x < \hat{l}_p(e, u)].$$

That is, x is not only an agreement point at stage s, but it has been so for every stage between t and s.

We now prove, by induction on e, that N_e and P_e are satisfied, and $A^{[e]}$ is recursive.

- *N_e is satisfied*
  $\hat{I}_e \leq_T A^{[<e]}$ as in the proof of the Thickness Lemma X.3.7, and $A^{[<e]}$ is recursive by induction hypothesis: since C is not recursive by hypothesis, $C \not\leq_T \hat{I}_e$, and thus N_e is satisfied by the Injury Lemma X.3.2.

- *P_e is satisfied*
  By induction hypothesis and the satisfaction of N_e, the Window Lemma X.3.5 gives $\liminf_{s\to\infty} \hat{R}(e, s) = \hat{R}(e) < \infty$. Suppose $A \simeq \{e\}^C$. Then $\lim_{s\to\infty} \hat{l}_p(e, s) = \infty$. Given x, recursively in A and C (and hence in C alone, since by the current hypothesis $A \leq_T C$) we can find a stage t_x after which $\hat{l}_p(e, s) > x$. But then for almost every x, i.e. for any x such that $\langle e, x, t_x\rangle > \hat{R}(e)$, we have

  $$x \in \mathcal{K} \ \Leftrightarrow\ \langle e, x, t_x\rangle \in A,$$

  and $\mathcal{K} \leq_T A \leq_T C$, which is a contradiction.

- *$A^{[e]}$ is recursive*
  Since $A \not\simeq \{e\}^C$, let z_e be the smallest z such that $A(z) \not\simeq \{e\}^C(z)$. Then $z_e = \liminf_{s\to\infty} \hat{l}_p(e, s)$.

  If $x \geq z_e$, then $\langle e, x, t\,\rangle$ can enter A only until a stage $s \geq t$ is reached such that $\hat{l}_p(e, s) = z_e$, and not afterwards. Thus, given x and t, find s as just stated. Then

  $$\langle e, x, t\,\rangle \in A \ \Rightarrow\ \langle e, x, t\,\rangle \in A_s.$$

  It is here that the new coding plays its essential role.

  If $x < z_e$, let s_0 be a stage such that $\mathcal{K}_{s_0}$, A_{s_0} and $\{e\}_{s_0}^{C_{s_0}}$ have all settled for every such x (note that $\{e\}^C$ converges on them), so that $\hat{l}_p(e, s) \geq z_e$

for every $s \geq s_0$. Given x and t, find (recursively) $s \geq s_0, t$ such that $\hat{R}(e,s) = \hat{R}(e)$. Then

$$\langle e, x, t\,\rangle \in A \;\Rightarrow\; \langle e, x, t\,\rangle \in A_{s+1}.$$

We thus have a procedure (depending on z_e, and hence nonuniform in e) to tell us whether $\langle e, x, t\,\rangle$ is in A, i.e. to decide $A^{[e]}$. $\square$

The coding technique used above was introduced by Sacks [1964a], to prove the Density Theorem. Before that, Sacks [1963a] had given a simpler (nonuniform) proof of the result (see X.6.14). The first uniform proof was given by Yates [1966a] using index sets (p. 633). There are thus three substantially different proofs of X.6.1.

Exercises X.6.2 Chains and antichains. a) *Every finite chain of r.e. degrees is extendable, and thus every maximal chain is countable.* (Hint: use the methods of X.6.1 and X.5.2.)

b) *Every finite antichain of nontrivial r.e. degrees is extendable, and thus every maximal antichain is countable.*

Density

We have seen in X.5.2 how the ideas used to obtain m-degrees incomparable with a given nontrivial one can be pushed to their limits, to prove the upward density of the r.e. m-degrees. In the case of r.e. Turing degrees we can actually obtain full density in a similar way.

Theorem X.6.3 Density Theorem for $\mathcal{R}$ (Sacks [1964a]) *Given r.e. degrees $\boldsymbol{b} < \boldsymbol{c}$, there is an r.e. degree $\boldsymbol{a}$ such that $\boldsymbol{b} < \boldsymbol{a} < \boldsymbol{c}$.*

Proof. Let B and C be r.e. sets such that $B <_T C$. We want to build an r.e. set $A \leq_T C$ such that

$$\begin{array}{lcl} P_{-1} & : & B \leq_T A \\ P_e & : & A \not\simeq \{e\}^B \\ N_e & : & C \not\simeq \{e\}^A. \end{array}$$

P_e and N_e are handled as in X.6.1 by the modified coding and the Sacks agreement methods, respectively. The only formal change is that, to satisfy P_e, we now try to code C into $A^{[e+1]}$ instead of into $A^{[e]}$. So $A^{[0]}$ is free, and we use it to code B (to satisfy P_{-1}, with the highest priority).

To achieve $A \leq_T C$ we use, as in the Strong Thickness Lemma X.3.9, the modified length of agreement function for negative requirements, and the associated restraint function:

$$\begin{aligned}\hat{m}(e,s) &= \max\{z : z \leq \max_{t\leq s} \hat{l}_n(e,t) \ \wedge\ (\forall y < z)(\{\hat{e}\}_s^{A_s}(y)\downarrow)\}\\ \hat{r}(e,s) &= \max\{\hat{u}(e,x,s) : x \leq \hat{m}(e,s)\}.\end{aligned}$$

The construction is as follows. We start with $A_0 = \emptyset$. At stage $s+1$, we put $\langle 0,x\rangle$ in A if $x \in B_s$, and we put $\langle e+1,x,t\,\rangle$ in A if

$$x \in C_s \ \wedge\ (\forall i \leq e)[\langle e+1,x,t\,\rangle > \hat{r}(i,s)] \ \wedge\ (\forall u)[t \leq u \leq s \ \Rightarrow\ x < \hat{l}_p(e,u)].$$

The proof that the requirements are satisfied is as in X.6.1. Since $A^{[0]} \equiv_T B$ and $A^{[e+1]}$ is recursive, we have $A^{[<e]} \leq_T B$ and so

$$\hat{I}_e \leq_T A^{[<e]} \leq_T B,$$

hence $C \not\leq_T \hat{I}_e$ because $C \not\leq_T B$. Thus N_e is satisfied, P_e is satisfied and $A^{[e+1]}$ is recursive, exactly as in X.6.1. The only formal change is that P_e is now satisfied because we try to code C into A, so that if $A \leq_T B$, then $C \leq_T A \leq_T B$, contradicting $B <_T C$.

Finally, $A \leq_T C$. For this we prove by induction that $A^{[e]} \leq_T C$ uniformly in e. First, $A^{[0]} \equiv_T B \leq_T C$. Suppose now that $A^{[i]} \leq_T C$ for every $i \leq e$, so that $A^{[<e+1]} \leq_T C$. Let $\langle e+1,x,t\,\rangle$ be given, and decide recursively in C whether $x \in C$. If $x \notin C$, then $\langle e+1,x,t\,\rangle \notin A$. If $x \in C$, find a stage s_0 such that $x \in C_{s_0}$. For any stage $s \geq s_0$ of nondeficiency for $A^{[<e+1]}$, the functions $\hat{r}(i,s)$ will drop back simultaneously for every $i \leq e$. Recursively in $A^{[<e+1]}$, and hence in C, find such an s: now

$$\langle e+1,x,t\,\rangle \in A \ \Leftrightarrow\ \langle e+1,x,t\,\rangle \in A_s.$$

Indeed, if $\langle e+1,x,t\,\rangle \notin A_s$ then it will never go into A, since the restraint functions $\hat{r}$ are minimal at s, and (as x in C_s) the only reason why $\langle e+1,x,t\,\rangle$ should not get into A is because, for some u between t and s, $x \geq \hat{l}_p(e,u)$. But then this will be true at any later stage as well. $\Box$

A different proof of the Density Theorem has been given by Yates [1966a], using index sets (see p. 633). A simpler, nonuniform proof in the style of X.6.14 for X.6.1 is not known.

Exercises X.6.4 (Robinson [1971]) a) *Given three r.e. degrees $\boldsymbol{b} < \boldsymbol{d} < \boldsymbol{c}$ there is an r.e. degree $\boldsymbol{a}$ incomparable with $\boldsymbol{d}$ and such that $\boldsymbol{b} < \boldsymbol{a} < \boldsymbol{c}$.* (Hint: combine the techniques of X.6.1 and X.6.3.)

b) *Given two r.e. degrees $\boldsymbol{b} < \boldsymbol{c}$, every countable partial ordering is embeddable in the r.e. degrees between $\boldsymbol{b}$ and $\boldsymbol{c}$.* (Hint: see X.1.9.)

Greatest lower bounds

As in the case of r.e. m-degrees, we now prove that g.l.b.'s of pairs of r.e. Turing degrees exist sometimes, but not always.

Theorem X.6.5 (Lachlan [1966b], Yates [1966]) *There exists a minimal pair of r.e. degrees.*

Proof. We want to build two r.e. sets A and B such that

$$\begin{array}{lcl} P_{2e} & : & \mathcal{W}_e \text{ infinite } \Rightarrow \mathcal{W}_e \cap A \neq \emptyset \\ P_{2e+1} & : & \mathcal{W}_e \text{ infinite } \Rightarrow \mathcal{W}_e \cap B \neq \emptyset \\ N_e & : & \{e\}^A \simeq \{e\}^B \simeq C \Rightarrow C \text{ recursive.} \end{array}$$

To satisfy N_e we use the same idea as in the proof of X.5.4 (to which we refer in the following), i.e. we try to preserve one side of the agreement. In particular, let

$$l(e,s) \ = \ \max\,\{z : (\forall y < z)[\{e\}_s^{A_s}(y) \simeq \{e\}_s^{B_s}(y)]\},$$

where both sides are convergent, and call s an ***e*-maximal stage** if

$$l(e,s) \geq \max_{t \leq s} \ l(e,t).$$

If we define the restraint function as in X.5.4, i.e.

$$r(e,s) = \begin{cases} 0 & \text{if } s \text{ is } e\text{-maximal} \\ \text{the greatest } e\text{-maximal stage } t < s & \text{otherwise,} \end{cases}$$

we have a problem. In X.5.4, to prove that $\lim_{s\to\infty} r(e,s) < \infty$ we used the fact that if $\{e\}^A$ and $\{e\}^B$ differ, then $\lim_{s\to\infty} l(e,s) < \infty$. As usual, this is not true for Turing reductions. However, we know (from Section 3) that to satisfy P_e we only need

$$\liminf_{s\to\infty} R(e,s) < \infty,$$

and this is what we will have to achieve. Note that, by defining r as above, we automatically have

$$\liminf_{s\to\infty} r(e,s) < \infty,$$

because if there are infinitely many e-maximal stages, then the lim inf is 0, while if there are only finitely many, then the lim inf is their greatest one. We are thus very near to the solution, and we only have to ensure that the restraint functions drop back simultaneously.

The solution consists of playing strategies one inside the other. For example, we know that $\liminf_{s\to\infty} r(0,s) < \infty$. Suppose we also knew its final value k:

then we could decide to disregard all stages in which $r(0,s) \neq k$, and play the strategy for N_1 only on the set

$$S_0 = \{s : r(0,s) = k\}.$$

The only thing we would have to do at stages which are not in S_0 would be to keep as good the restraints imposed at the last stage in S_0 (to avoid ruining, at intermediate stages, computations that we want to preserve at true stages).

The problem is, of course, that we do not know what k is going to be. But at stage s we do have a value $r(0,s)$, and we can decide to act as if this were the real $\liminf_{s\to\infty} r(0,s)$, i.e. we consider the sets

$$S_0^k = \{s : r(0,s) = k\}$$

and play the strategy above.

We still have to coordinate different strategies. We cannot simply disregard the strategies for $k' < k$, since one such k' could be the real lim inf. Thus we also keep the restraints relative to $k' < k$. If k is going to be the real lim inf, this will not hurt, since $S_0^{k'}$ will be finite for each $k' < k$, by definition.

We can then define $R(e,s)$ directly, by induction on e:

- $R(0,s) = r(0,s)$
- $R(e+1,s)$ is the maximum of:
  1. $R(e,s)$
  2. $r(e,t)$, for any $t < s$ such that $R(e,t) < R(e,s)$
  3. the greatest $t < s$ which is $e+1$-maximal, if s is not

where now s is an $\boldsymbol{e+1}$**-maximal stage** if

$$l(e+1,s) \geq \max\,\{l(e+1,t) : t < s \;\wedge\; R(e,t) = R(e,s)\}.$$

1 simply gives the sum of the restraints for negative requirements of higher priority; 2 takes into account the possibility that $R(e,s)$ is not the lim inf; 3 is the usual restraint, with the appropriate notion of maximality (i.e. restricted to the stages at which the previous restraints look the same).

The construction is now exactly as in X.5.4. Start with $A_0 = B_0 = \emptyset$. At stage $s+1$, consider the least positive requirement with index $\leq s$ which has not yet been satisfied and can be satisfied now, and satisfy it. More precisely, let $e \leq s$ be the smallest number such that:

- $\mathcal{W}_{e,s} \cap A_s = \emptyset$ or $\mathcal{W}_{e,s} \cap B_s = \emptyset$
- for some x, $x \in \mathcal{W}_{e,s} \;\wedge\; x \geq 2e \;\wedge\; (\forall i \leq e)[x > r(i,s)]$

and put one such x in A if $\mathcal{W}_{e,s} \cap A_s = \emptyset$, and in B otherwise.

- $\lim_{s\to\infty} R(e,s) < \infty$, *for every* e
  By induction on e. For $e = 0$ we know this already, since $R(0,s) = r(0,s)$. Now let $\liminf_{s\to\infty} R(e,s) = k$ and $S_e = \{s : R(e,s) = k\}$. By definition of lim inf, there are only finitely many t such that $R(e,t) < k$: then clause 2 of the definition of R only contributes a finite amount. For $s \in S_e$, $R(e,s) = k$ is constant. If there are infinitely many $e+1$-maximal stages $s \in S_e$, then $\liminf_{s\to\infty} R(e+1,s) < \infty$ because clause 3 is not applied infinitely often; if there are only finitely many, then clause 3 only contributes a finite amount.

- N_e *is satisfied, for every* e
  Let $k = \liminf_{s\to\infty} R(e-1,s)$, and $S = \{s : R(e-1,s) = k\}$ (notice that S is recursive). Choose s_0 such that, for every $s \geq s_0$, N_e is not injured at s and $R(e-1,s) \geq k$ (which is possible, by definition of lim inf). Suppose that
  $$\{e\}^A \simeq \{e\}^B \simeq C.$$
  C is recursive because it is enough, given x, to find $s > s_0$ such that $s \in S$, $l(e,s) > x$, and s is e-maximal (on S). Then
  $$\{e\}_s^{A_s}(x) \simeq C(x),$$
  since one side of the agreement is always preserved on S by definition, and outside S by clause 2 above (even when we consider $t \notin S$ with $R(e-1,t) > k$ we preserve the restraint for k, and after s_0 we always have $R(e-1,s) \geq k$).

- P_e *is satisfied, for every* e
  As usual. $\square$

The proof presented above is due to Lachlan [1973], and it shows in a simple but nontrivial case the technique of having different strategies for a single requirement (discussed in Section 4). Here there is only one scheme of strategies for each single requirement, and the combination of strategies is a simple nesting.

The positive requirements of the proof above are finitary, hence the negative requirements can be injured only finitely often. However, the restraint functions may be unbounded, and thus *the argument is intermediate between finite and infinite injury*. In the terminology of Section 3, we need a Window Lemma but not an Injury Lemma. The next exercises sketch proofs by the pinball machine and the tree methods.

Exercises X.6.6 a) *Build a minimal pair of r.e. degrees by using the pinball machine method.* (Lerman [1973]) (Hint: as in the setting of X.3.6, the holes eject elements that can satisfy their associated positive requirements, and the gates open when the restraints of their associated negative requirements drop back. The restraint functions $r(e, s)$ are defined as in the previous proof, using e-maximal stages, but the modified restraint functions $R(e, s)$ are not needed because we are not trying to ensure that the $r(e, s)$ drop back simultaneously. Rather, at stage $s + 1$ we make sure that: on the one hand, for each positive requirement with index $\leq s$ which has not yet been satisfied and can be satisfied, the associated hole ejects elements that can satisfy it; and, on the other hand, the smallest element ejected for the positive requirement of highest priority that has not yet been satisfied, and that can pass through some gates that are open for it at that stage, does so, and stops in front of the first gate that is closed for it.)

b) *Build a minimal pair of r.e. degrees by using the tree method.* (Hint: as in the setting of X.4.3, we code partial information about the outcomes of positive or negative requirements by strings σ, with the conventions that $\sigma(2e)$ is an integer guessing the value of $\liminf_{s\to\infty} r(e, s)$, and $\sigma(2e+1)$ is 0 or 1 according to whether the positive requirement P_e has been satisfied or not. Once again the restraint functions $r(e, s)$ are defined as in the previous proof, using e-maximal stages, and the modified restrain functions $R(e, s)$ are not needed. At stage $s+1$ we consider the unique string σ_s of length s that looks correct at stage s, let $e \leq s$ be the smallest number such that:

- $\mathcal{W}_{e,s} \cap A_s = \emptyset$ or $\mathcal{W}_{e,s} \cap B_s = \emptyset$
- for some x, $x \in \mathcal{W}_{e,s} \ \wedge \ x \geq 2e \ \wedge \ (\forall i \leq e)(\forall t \leq s)(\sigma_t \preceq \sigma_s \ \Rightarrow \ x > \sigma_t(2i))$

and put one such x in A if $\mathcal{W}_{e,s} \cap A_s = \emptyset$, and in B otherwise (notice that the consideration of strings $\sigma_t \prec \sigma_s$ corresponds to clause 2 of the definition of $R(e+1, s)$ in the proof of X.6.5). Then R_σ is satisfied for each σ on the true path, because the restraints of all negative requirements simultaneously drop back to their lim inf on the true path.)

Exercises X.6.7 a) *There is a minimal pair of low r.e. degrees.* (Lachlan [1966b], Yates [1966]) (Hint: this is automatic in the proof of X.6.5, see e.g. X.1.5.)

b) *There is a minimal pair of high r.e. degrees.* (Lachlan [1966b]) (Hint: to obtain A and B of high degree use the methods of XI.1.13, namely build them as thick subsets of appropriate sets. One now has to guess also whether the columns are finite or not, and if so what is their cardinality. An Injury Lemma is now needed, but since the columns are always recursive, so are the injury sets. The argument can thus be put into any of the frameworks of Sections 3 and 4.)

Cooper [1974] has proved that *every high r.e. degree bounds an r.e. minimal pair*. This result is the best possible, since Downey, Lempp and Shore [1993] have proved that *there is a high$_2$ r.e. degree that does not bound r.e. minimal pairs*. The first example of an r.e. degree not bounding r.e. minimal pairs was given by Lachlan [1979].

In the terminology of X.6.9, Ambos-Spies, Jockusch, Shore and Soare [1984] have proved that *every noncappable r.e. degree bounds an r.e. minimal pair*. Thus an r.e. degree that does not bound r.e. minimal pairs must be cappable.

Exercises X.6.8 Branching r.e. degrees. An r.e. degree is called **(non)branching** if it is (not) the g.l.b. of two incomparable r.e. degrees above it.

a) **Branching Theorem for $\boldsymbol{\mathcal{R}_{wtt}}$** (Cohen [1975]) *Every wtt-incomplete r.e. wtt-degree is branching.* (Hint: given $C <_{wtt} \mathcal{K}$, build A and B such that $C <_{wtt} A, B$ by the methods of X.6.1, and such that if $D \leq_{wtt} A, B$ then $D \leq_{wtt} C$ by the methods of X.6.5.)

b) *There exists a nonzero branching degree.* (Lachlan [1966b]) (Hint: as in the minimal pair construction, build a nonrecursive r.e. set A and two r.e. sets B and C not recursive in A and such that

$$\{e\}^{A \oplus B} \simeq \{e\}^{A \oplus C} \simeq D \;\Rightarrow\; D \leq_T A.)$$

c) **Non-Branching Theorem for $\boldsymbol{\mathcal{R}}$** (Lachlan [1966b]) *There exists an incomplete nonbranching degree.* (Hint: the negative requirements are

$$N_e \;:\; \mathcal{K} \not\simeq \{e\}^A,$$

and are satisfied by the Sacks agreement method, with the usual restraint effect. For any i and j, if $A <_T \mathcal{W}_i, \mathcal{W}_j$ we do not want A to be the g.l.b. of $\mathcal{W}_i$ and $\mathcal{W}_j$, and we build $B_{i,j} \leq_T \mathcal{W}_i, \mathcal{W}_j$ such that $B_{i,j} \not\leq_T A$, by satisfying the requirements

$$R_{i,j,e} \;:\; A <_T \mathcal{W}_i, \mathcal{W}_j \;\Rightarrow\; B_{i,j} \not\simeq \{e\}^A.$$

They are satisfied by the usual strategy of choosing a witness $x \notin B_{i,j}$, waiting for $\{e\}^A(x)$ to converge, and then putting x in $B_{i,j}$ if $\{e\}^A(x) \simeq 0$.

To obtain $B_{i,j} \leq_T \mathcal{W}_i, \mathcal{W}_j$ one would like to use permitting, but while this works separately for $\mathcal{W}_i$ and $\mathcal{W}_j$ if they are strictly above A, since in particular they are not recursive, there is no reason to think that a witness will ever be permitted by both $\mathcal{W}_i$ and $\mathcal{W}_j$ simultaneously. We can use permitting to have $B_{i,j} \leq_T \mathcal{W}_j$, and play with A to ensure $B_{i,j} \leq_T \mathcal{W}_i$. Namely, we associate a_x with x and if we ever put x into $B_{i,j}$, it will be at a stage in which a_x goes into A. This makes $B_{i,j}$ recursive in A and a_x (as a function of x). To ensure $B_{i,j} \leq_T \mathcal{W}_i$, it is then enough to obtain a_x recursively in $\mathcal{W}_i$, since $A \leq_T \mathcal{W}_i$. This is easy to achieve, as follows: we obtain a_x as the limit of a_x^s, and move a_x^s (to satisfy negative requirements of higher priority) only if x is permitted by $\mathcal{W}_i$, i.e.

$$a_x^{s+1} \neq a_x^s \;\Rightarrow\; (\exists z \leq x)(z \in \mathcal{W}_{i,s+1} - \mathcal{W}_{i,s}).$$

We thus satisfy $R_{i,j,e}$ as follows: wait until a stage s is found such that, for some fresh witness x, x is permitted by $\mathcal{W}_i$ and $\{e\}_s^{A_s}(x)$ converges. Move a_x^s to a position greater than the restraints of higher priority and the use function of $\{e\}_s^{A_s}(x)$, so that by putting a_x into A we will not injure the computation. Then wait until $t > s$ is found, such that x is permitted by $\mathcal{W}_j$ and $\{e\}_s^{A_s}(x)$ has not been injured. Then put

a_x into A (always), and put x into $B_{i,j}$ if and only if $\{e\}_t^{A_t}(x) \simeq 0$. The hypothesis $A <_T \mathcal{W}_i, \mathcal{W}_j$ is used to show that each permitting works.)

d) *Below any nonzero r.e. degree there exists a nonbranching degree.* (Lachlan [1966b]) (Hint: add permitting to the construction of part c).)

Structural results for the (non)branching degrees are in Fejer [1982], [1983] and Slaman [1991]. In particular, *both the branching and the nonbranching degrees are dense in the r.e. degrees.*

Exercises X.6.9 Cappable degrees. A degree is called **(non)cappable** if it is (not) one half of a minimal pair.

a) **Non-Capping Theorem for $\mathcal{R}$** (Yates [1966]) *There exists a nontrivial noncappable degree.* (Hint: an indirect proof is given in X.6.24. For a direct proof, since $A^{[e]} \leq_T A$ automatically holds, it is enough to ensure that whenever $\mathcal{W}_e$ is nonrecursive, $A^{[e]} \leq_T \mathcal{W}_e$ (by permitting) and $A^{[e]}$ is not recursive. Thus $A^{[e]}$ witnesses that A and $\mathcal{W}_e$ do not form a minimal pair. The requirements are:

$$\begin{array}{lcl} R_{e,i} & : & \mathcal{W}_e \text{ nonrecursive} \Rightarrow A^{[e]} \leq_T \mathcal{W}_e \wedge A^{[e]} \neq \overline{\mathcal{W}_i} \\ N_e & : & \mathcal{K} \not\simeq \{e\}^A, \end{array}$$

and they can be satisfied in a straightforward way.)

b) *The cappable degrees together with* $\mathbf{0}$ *are closed downward* (they are actually an ideal, by Ambos-Spies, Jockusch, Shore and Soare [1984]) *and the noncappable degrees together with* $\mathbf{0}'$ *are a filter.* (Hint: closure upward is trivial. Let $\boldsymbol{a_0}$ and $\boldsymbol{a_1}$ be noncappable and suppose $\boldsymbol{a_0} \cap \boldsymbol{a_1}$ exists. If $\boldsymbol{b}$ is a nontrivial r.e. degree, there are nonzero r.e. degrees $\boldsymbol{c_0}$ and $\boldsymbol{c_1}$ such that $\boldsymbol{c_0} \leq \boldsymbol{b}, \boldsymbol{a_0}$ and $\boldsymbol{c_1} \leq \boldsymbol{c_0}, \boldsymbol{a_1}$. Thus $\boldsymbol{c_1} \leq \boldsymbol{b}, \boldsymbol{a_0} \cap \boldsymbol{a_1}$ and $\boldsymbol{a_0} \cap \boldsymbol{a_1}$ is not cappable.)

c) *If A is coinfinite, has the splitting property and is not hyperhypersimple, then its degree is noncappable.* (Maass, Shore and Stob [1981]) (Hint: let f witness that A is not hyperhypersimple. We may suppose that $\mathcal{W}_{f(e)}$ has no elements less than e. Given an r.e. set C, the r.e. set defined as follows

$$x \in B \Leftrightarrow (\exists e)(\exists s)(x \in \mathcal{W}_{f(e),s} - A_s \wedge e \in C_{s+1} - C_s)$$

is such that $A \cap B \leq_T A$, $B \leq_T C$, and if B is recursive, then $C \leq_T A$. If A has the splitting property, let $C \not\leq_T A$. To show that A and C do not form a minimal pair, take B as above, which is then nonrecursive. Split B into B_0 and B_1 nonrecursive and such that $B_0 \subseteq A$. Then $B_0 \leq_T C$ because $B_0 \leq_T B \leq_T C$, and $B_0 \leq_T A$ because $A \cap B \leq_T A$.)

d) *The degrees of nonhyperhypersimple sets having the splitting property nontrivially split every jump class.* (Maass, Shore and Stob [1981]). (Hint: let S be the set of degrees of nonhyperhypersimple sets having the splitting property. First, each jump class intersects $\overline{S}$, because there are high minimal pairs by X.6.7.b, and the cappable degrees are downward closed. Second, S contains a low degree, because there are low promptly simple sets, which have the splitting property by IX.2.6, but are not hyperhypersimple by IX.2.25.a. Third, every jump class different from $\mathbf{H}_1$ intersects S, because the r.e. sets having the splitting property form a filter in $\mathcal{E}$ by IX.2.6.f, and *if A is not hyperhypersimple, then every degree above the degree of A can be represented by some r.e. set $B \supseteq A$* (Lachlan [1968d]). To prove the latter assertion, given $A \leq_T C$ it is enough to let, in the notation of part c),

$$B = A \cup \{x : (\exists e)(\exists s)(x \in \mathcal{W}_{f(e),s} \ \wedge \ e \in C_{s+1} - C_s)\}.$$

Finally, there are high degrees in S because it is easy to build an r-maximal, nonhyperhypersimple, promptly simple set.)

Ambos-Spies, Jockusch, Shore and Soare [1984] have proved the converse of part c), i.e. any noncappable degree contains a nonhyperhypersimple set having the splitting property, and it thus follows that *the noncappable degrees form a definable filter which nontrivially splits all jump classes*. In the same paper, it is also proved that *the noncappable degrees are exactly those containing promptly simple sets*, and *the cappable degrees form an ideal, which is also definable and nontrivially splits all jump classes*. One thus has a definable decomposition of the r.e. degrees into an ideal and a filter. The quotient structure of the r.e. degrees modulo this ideal has been studied by Schwarz [1984], Yi [1996] and Sui and Zhang [199?].

The **Non-Capping Theorem for $\mathcal{R}_{wtt}$** holds by the proof of part a). Ambos-Spies [1985c] has proved that *the T-degree of an r.e. set is noncappable if and only if its wtt-degree is noncappable*, so that most results on (non)cappability in the r.e. T-degrees carry over to the r.e. wtt-degrees.

The proof of the next result is not an extension of the proof given in X.5.6 for the r.e. m-degrees, but a simpler one.

Theorem X.6.10 (Lachlan [1966b], Yates [1966]) *There are two r.e. degrees without g.l.b.*

Proof. We want to build two r.e. sets A and B such that whenever $\mathcal{W}_e \leq_T A, B$ there is an r.e. set $V_e \leq_T A, B$ such that $V_e \not\leq_T \mathcal{W}_e$, so that $\mathcal{W}_e$ is not the g.l.b. of A and B (notice that A and B will automatically be nonrecursive). We will construct the r.e. sets V_e, satisfying the requirements:

$$R_{e,i} \ : \ \{(e)_1\}^A \simeq \{(e)_2\}^B \simeq \mathcal{W}_{(e)_3} \ \Rightarrow \ V_e \not\simeq \{i\}^{\mathcal{W}_{(e)_3}}.$$

To satisfy $V_e \not\simeq \{i\}^{\mathcal{W}_{(e)_3}}$ we adopt the usual strategy: choose $z \notin V_{e,s}$, wait until $\{i\}^{\mathcal{W}_{(e)_3}}(z) \simeq 0$, then put z into V_e and restrain the elements used negatively in the computation. Note that $\mathcal{W}_{(e)_3}$ is given, so we cannot directly restrain elements from entering it. But we can control A and B, and we are interested only in the case when $\{(e)_1\}^A \simeq \{(e)_2\}^B \simeq \mathcal{W}_{(e)_3}$. We thus apply the previous strategy only when every element used negatively in the computation $\{i\}^{\mathcal{W}_{(e)_3}}(z)$ is obtained via $\{(e)_2\}^B$. Then, instead of restraining elements from entering $\mathcal{W}_{(e)_3}$, we can restrain elements from entering B (by preserving those computations), with the same effect.

We also want $V_e \leq_T A, B$, and we thus decide to put into both A and B every element that goes into V_e, to be able to recover V_e recursively from A and B. The problem is that, by putting z into B, we might destroy the computation $\{i\}^{\mathcal{W}_{(e)_3}}(z)$. We thus modify the construction as follows. When $\{i\}^{\mathcal{W}_{(e)_3}}(z) \simeq 0$ and B controls the computation, put z into A and appropriately restrain B (i.e. do not change any value of B which produce - via $\{(e)_2\}^B$ - elements used negatively in the computation $\{i\}^{\mathcal{W}_{(e)_3}}(z)$). Wait for a later stage in which the computation may be controlled through A via $\{(e)_1\}^A$ (if this never comes, then $\{(e)_1\}^A \not\simeq \{(e)_2\}^B$ and $R_{e,i}$ is vacuously satisfied). Then drop the restraint on B, put z in both V_e and B, and restrain A to preserve the computation $\{i\}^{\mathcal{W}_{(e)_3}}(z)$.

Now $V_e \leq_T B$ by construction, since z goes into V_e at the same stage it goes into B. Moreover, if $\{(e)_1\}^A \simeq \{(e)_2\}^B$, then we can prove $V_e \leq_T A$ as follows. If z goes into A at a stage s_z (which we can find recursively once we know, recursively in A, whether $z \in A$), it is enough to search for a stage $t_z > s_z$ such that either some requirement of higher priority is satisfied and causes z to be dropped as a witness (in which case z never goes into V_e), or z goes into V_e at stage t_z. Note that such a stage must exist, because we are supposing $\{(e)_1\}^A \simeq \{(e)_2\}^B$. $\Box$

To obtain $V_e \leq_T B$ we used the usual permitting, i.e. we put x into V_e only when it goes into B (so x permits itself). To obtain $V_e \leq_T A$ we instead used the **delayed permitting** introduced by Lachlan [1966b], i.e. we put x into V_e somewhat after it has already gone into A (so x is not permitted in the usual sense), and still succeed because there is a function recursive in A that bounds the stages in which x can enter V_e.

Corollary X.6.11 $\boldsymbol{\mathcal{R}}$ *is not a lattice.*

Lachlan [1966b] obtained the corollary as a consequence of the Non-Diamond Theorem (see X.6.26.a). Yates [1966] constructed an infinite ascending sequence of r.e. degrees and an r.e. exact pair for it. The proof above, due to Jockusch [1981a], combines in a rudimentary form the ideas from both the original proofs, namely delayed permitting (from the Non-Diamond Theorem) and the technique of holding one side or another of the computation (from the minimal pair construction).

Since permitting also works for *wtt*-reducibility (see p. I.338), the same proof shows that $\boldsymbol{\mathcal{R}_{wtt}}$ *is not a lattice* (Ladner and Sasso [1975]).

Exercises X.6.12 a) *Two r.e. degrees have g.l.b. in the r.e. degrees if and only if they have g.l.b. in the degrees, and the g.l.b.'s coincide when they exist.* (Lachlan [1966b]) (Hint: we show that if A and B are r.e. and $D \leq_T A, B$, then there is C r.e. such that $D \leq_T C \leq_T A, B$. Let $\{e\}^A \simeq \{i\}^B \simeq D$. Given x, let C_x consist of the stages $\leq t$ such that an agreement $\{e\}_t^{A_t}(x) \simeq \{i\}_t^{B_t}(x)$ is later replaced by a new agreement with different value, and let $C = \bigoplus_{x \in \omega} C_x$.

To see that $D \leq_T C$, notice that if $s \notin C_x$ and $\{e\}_s^{A_s}(x) \simeq \{i\}_s^{B_s}(x)$, then this is the final value. To see that $C \leq_T A$, let x be given. Look for a stage s such that $\{e\}_s^{A_s}(x)\downarrow$ and A has settled on the use of the computation. Then the value $\{e\}_s^{A_s}(x)$ is final and no agreement with $\{i\}^B(x)$ reached after s can later be replaced by one with a different value. Similarly for $C \leq_T B$.)

b) *There are r.e. sets with g.l.b. in the r.e. wtt-degrees but not in the (r.e.) Turing degrees.* (Hint: let A and B form a minimal pair and C be of nonbranching degree below A, which exists by X.6.8.d. Then A and $B \oplus C$ have no g.l.b. in the Turing degrees. Moreover, if $D \leq_{wtt} A, B \oplus C$ then $D \leq_{wtt} C$ by distributivity X.6.27.c, and the fact that A and B form a minimal pair: thus C is the g.l.b. of A and $B \oplus C$ in the *wtt*-degrees.)

The analogue of part a) also holds for the r.e. *wtt*-degrees, by a similar proof. Fejer and Shore [1988] show that it fails for the r.e. *tt*-degrees. The problem does not arise for r.e. *m*-degrees, since they are an ideal of the *m*-degrees; similarly for the r.e. 1-degrees.

Downey and Stob [1986] show that the opposite of b) can also happen.

Ambos-Spies [1984a], [1984b] has studied the pairs of r.e. degrees without g.l.b., proving in particular that *no nontrivial interval of the r.e. degrees is a lattice.*

On the other hand, Fischer [1986] has proved that *some nontrivial lower cone of the r.e. wtt-degrees is a lattice, but no nontrivial upper cone is.* More results on the g.l.b.'s of r.e. *wtt*-degrees are in Fischer [1986], Downey and Stob [1986], Sui [1987], Downey [1989a], [1989b], Jiang and Sui [1995].

Least upper bounds

Least upper bounds of r.e. degrees are better behaved than greatest lower bounds. First, *for any pair* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *of r.e. degrees, their l.u.b. in the r.e. degrees always exists and it coincides with the l.u.b.* $\boldsymbol{a} \cup \boldsymbol{b}$ *in the degrees.* Second, the next result shows that there is no analogue of nonbranching degrees (the analogue of minimal pairs was already automatically obtained in X.1.5).

Theorem X.6.13 Splitting Theorem for $\mathcal{R}$ (Sacks [1963a]) *Every nonzero r.e. degree is the l.u.b. of two incomparable r.e. degrees.*

Proof. Given a nonrecursive r.e. set C, we want to build disjoint r.e. sets A and B such that $A \cup B = C$ and $C \not\leq_T A, B$. Then the degree of C is the l.u.b. of the degrees of A and B:

1. $A \leq_T C$ and $B \leq_T C$
   Given x, first see if it is in C. If not, then it is neither in A nor in B.

Otherwise, generate A and B, until x appears in one of them (which it does, since C is their union). Then x is in the first set in which it appears, and not in the other (since A and B are disjoint).

2. $A \oplus B \equiv_T C$
$A \oplus B \leq_T C$ by 1. The converse is trivial, since

$$x \in C \ \Leftrightarrow\ x \in A \cup B \ \Leftrightarrow\ 2x \in A \oplus B \ \vee\ 2x+1 \in A \oplus B.$$

3. A and B are Turing incomparable
If $A \leq_T B$, then $A \oplus B \equiv_T B$ and, by 2, $B \equiv_T C$. But by construction $C \not\leq_T B$. Similarly if $B \leq_T A$.

The requirements for the construction of A and B are the following:

$$\begin{array}{lcl} P & : & x \in C_{s+1} - C_s \ \Rightarrow\ x \in A_{s+1} \text{ or } x \in B_{s+1} \\ N_e^A & : & C \not\simeq \{e\}^A \\ N_e^B & : & C \not\simeq \{e\}^B. \end{array}$$

The strategy for the negative requirements is the Sacks agreement method, which produces restraints $r^A(e,s)$ and $r^B(e,s)$. Since P ensures that $C = A \cup B$, it has the highest priority. We can thus consider, for example, the following priority list:

$$P > N_0^A > N_0^B > N_1^A > N_1^B > \cdots$$

The construction is as follows, given an enumeration of C producing exactly one element at each stage. We start with $A_0 = B_0 = \emptyset$. At stage $s+1$, let $x \in C_{s+1} - C_s$ be the unique element generated by C. We try to preserve the requirements of higher priority, so let e be the least one (if any) such that

$$x \leq r^A(e,s) \text{ or } x \leq r^B(e,s).$$

If there is no such e, we do not injure any condition by putting x into A or B, and we thus put it, for example, in A. Otherwise, we put x in B if $x \leq r^A(e,s)$, and in A otherwise.

Notice that, for any s, only finitely many of the restraint functions are different from 0, since (by definition, see II.3.12) a computation $\{e\}_s^{A_s}(y)$ or $\{e\}_s^{B_s}(y)$ converges only if $e \leq s$. Thus the construction is recursive, and A and B are r.e. Also, the construction makes them disjoint, and satisfies P (which has highest priority).

Using the fact that C is nonrecursive, we can show that the negative requirements are satisfied as in X.1.10, once we know that they are injured only finitely often. But this cannot be proved independently for each requirement (as in X.1.10), since P is infinitary. We thus proceed by induction on the priority ordering.

N_0^A, having highest priority (among the negative requirements), is never injured (i.e. we put in B every element that could injure it). Thus it is satisfied, and $\lim_{s\to\infty} r^A(0,s) < \infty$ as in X.1.10. Let s_0 be a stage after which $r^A(0,s)$ is constant, and every element of C less than $r^A(0,s_0)$ has been generated in C_{s_0}. N_0^B has highest priority after s_0, and is no longer injured (i.e. we put in A every element that could injure it). Thus it is satisfied, and $\lim_{s\to\infty} r^B(0,s) < \infty$. One can then proceed by induction. $\square$

Corollary X.6.14 *For every r.e. degree* $\mathbf{c}$ *such that* $\mathbf{0} < \mathbf{c} < \mathbf{0}'$*, there is an incomparable r.e. degree* $\boldsymbol{a}$*.*

Proof. Let C be an r.e. nonrecursive and incomplete set. Split $\mathcal{K}$ into A and B such that $C \not\leq_T A, B$, as in the proof above. It cannot be both $A \leq_T C$ and $B \leq_T C$, otherwise

$$\mathcal{K} = A \cup B \equiv_T A \oplus B \leq_T C,$$

and C would be complete, contradicting the hypothesis. Thus one of A and B is not recursive in C, and hence is incomparable with it because C is not recursive in any of them. $\square$

Sacks originally invented the agreement method to prove the Splitting Theorem. This is a strong result because, besides the corollary (proved in X.6.1 with a more difficult but uniform proof), it implies the Friedberg-Muchnik Splitting Theorem (obtained in IX.2.4 by a priority argument without injuries), the nonexistence of r.e. minimal degrees (obtained in X.2.8 by permitting), and the existence of incomparable r.e. degrees below any nonzero r.e. degree (X.2.8).

The proof just given is *a finite injury argument with some peculiar features, due to the infinitary nature of the (single) positive requirement*. We are able to injure the negative requirements at most finitely often because we play simultaneously with two sets, but we are not able to recursively control the number of injuries as we did in the arguments of Sections 1 and 2 (see X.6.15.b). Also, we prove by induction that the negative requirements are satisfied, as in infinite injury arguments (and not independently for each e, as in usual finite injury arguments).

It is the failure of the simple fact (proved at the beginning of the proof) that if A and B are disjoint r.e. sets, then $A \cup B \equiv_T A \oplus B$, that does not allow us to apply the same proof to all strong reducibilities. For example, while $A \leq_{tt} A \oplus B$ always holds, $A \leq_{tt} A \cup B$ may fail for disjoint r.e. sets A and B (as proved in III.3.17). However, the same proof as above gives $A \cup B \equiv_{wtt} A \oplus B$, and thus proves the **Splitting Theorem for $\boldsymbol{\mathcal{R}_{wtt}}$**. Actually, the proof shows that $A \cup B \leq_{btt} A \oplus B$ for disjoint r.e. sets, and since $A \oplus B \leq_m \mathcal{K}$ it follows that *for any reducibility* $\leq_r$ *between* $\leq_{btt}$ *and* $\leq_T$*,* $\mathbf{0}'_r$ *splits into two incomparable r.e.* r*-degrees* (this fails for m-reducibility, see X.5.9).

Exercises X.6.15 a) *Every nonzero r.e. degree is the l.u.b. of two incomparable low r.e. degrees.* (Sacks [1963]) (Hint: as in X.1.5, this is automatically obtained in the proof of X.6.13; alternatively, one can easily add requirements for lowness.)

b) *There is in general no recursive bound of the number of injuries to negative requirements in X.6.13.* (Hint: build an r.e. set A such that, for any e, A is generated in such a way as to injure the requirement N_e more than $\varphi_e(e)$ times, if this converges. Use the Fixed-Point Theorem.)

After obtaining the Splitting Theorem, Sacks tried to prove the Density Theorem by showing that, given two r.e. degrees $\boldsymbol{d} < \boldsymbol{c}$, there are incomparable r.e. degrees $\boldsymbol{a}$ and $\boldsymbol{b}$ between $\boldsymbol{d}$ and $\boldsymbol{c}$ such that $\boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{c}$, but was unable to do so. He then devised the coding method, with which he could prove the Density Theorem directly. The question remained of whether Splitting and Density could be combined. The first exercise below shows that this is so for the r.e. *wtt*-degrees, while the fourth shows that in *special cases* the result does hold for Turing degrees as well. The second exercise proves that the *strategy* used for *wtt*-degrees fails to extend to Turing degrees, while the third shows that the *method* (i.e. splitting sets) fails in general. Lachlan [1975a] proved that the *result* itself fails (historically, the first application of the **monster injury priority argument**). Harrington then strengthened this by showing that $\mathbf{0}'$ itself cannot always be split above a lower degree (an easy proof of this improvement, given the original result of Lachlan, is in Jockusch and Shore [1983]).

Exercises X.6.16 Splitting and Density combined.

a) **Splitting and Density Theorem for $\boldsymbol{\mathcal{R}_{wtt}}$** (Ladner and Sasso [1975]) *Splitting and density can be combined for the r.e. wtt-degrees.* (Hint: Given $D <_{wtt} C$, we want A and B disjoint and such that $C = A \cup B$, $C \not\leq_{wtt} A \oplus D, B \oplus D$. The requirements are:

$$\begin{array}{lcl} P & : & x \in C_{s+1} - C_s \ \Rightarrow\ x \in A_{s+1} \text{ or } x \in B_{s+1} \\ N_e^A & : & C \not\simeq \{(e)_1\}^{A \oplus D} \text{ with bound } \varphi_{(e)_2} \\ N_e^B & : & C \not\simeq \{(e)_1\}^{B \oplus D} \text{ with bound } \varphi_{(e)_2}. \end{array}$$

As usual, let e.g.

$$\begin{array}{lcl} l^A(e,s) & = & \max \{z : (\forall y < z)[C_s(y) \simeq \{(e)_1\}_s^{A_s \oplus D_s}(y)]\} \\ r^A(e,s) & = & \max \{u((e)_1, x, s) : x \leq \max_{t \leq s} l^A(e,t)\}, \end{array}$$

i.e. we use a modified length of agreement function as in X.3.9 or X.6.3, and the restraints are nondecreasing. The construction is as in X.6.13. N_0^A is never injured. Suppose it is not satisfied. Then $\lim_{s\to\infty} l^A(0,s) = \infty$. Given x, find the first stage s such that $l^A(0,s) > x$ and D has settled on every element up to $\varphi_{(0)_2}(x)$. Then $C_s(x) = C(x)$ because A never injures N_0^A, D does not injure it after stage s, and if

C changes, then disagreement is preserved. Thus $C \leq_{wtt} D$ with bound $\varphi_{(0)_2}$, which is a contradiction. Then N_0^A is satisfied, $\lim_{s\to\infty} r^A(0,s) < \infty$.)

b) *The strategy of part a) fails for Turing degrees.* (Hint: N_0^A is satisfied as above. Otherwise, $\lim_{s\to\infty} l^A(0,s) = \infty$ and $\{0\}^{A\oplus D}$ is total. Given x, find s such that $l^A(0,s) > x$. From then on A no longer interferes with the computation, so that $\{0\}^{A\oplus D}(x) \simeq \{0\}^{A_s\oplus D}(x)$. Compute the latter recursively in D, and obtain $t > s$ at which D has settled on the elements used in the computation. From then on, D also no longer interferes, so that $C_t(x) = C(x)$ and $C \leq_T D$, which is a contradiction. But it might be that $\lim_{s\to\infty} r^A(0,s) = \infty$, if $\{0\}^{A\oplus D}$ diverges for some argument with an unbounded use function. This can happen even if N_0^A is never injured by A, because D is given in advance. And if it does happen, then A is recursive. Indeed, x can enter A only when it is not restrained and $r^A(0,s)$ increases, and thus for any stage s such that $x < r^A(0,s)$ one has $x \in A$ if and only if $x \in A_{s+1}$. In other words, *the splitting construction is not compatible with restraints going to infinity.*)

c) *There is an r.e. Turing complete set A, and an r.e. set $B <_T A$, such that A cannot be split in the disjoint union of two r.e. sets in which B is recursive.* (Cooper) (Hint: for any A the index sets I of the r.e. sets recursive in some pair of r.e. sets forming a disjoint union of A is Σ_4^0. If A is a T-complete nonmitotic set (X.6.18.c) and B as stated does not exist, then I is the index set of the T-incomplete r.e. sets, which is $\Pi_4^0 - \Sigma_4^0$ by X.9.16.)

d) *Splitting and density can be combined for $\boldsymbol{\mathcal{R}}$, above low degrees.* (Robinson [1971]) (Hint: the problem in part b) arises from the fact that, for example, N_e^A might be satisfied with $\{e\}^{A\oplus D}$ undefined on the least disagreement, thus pushing the restraint function to infinity. But if D is low, then the question of whether $\{e\}^D(x)$ converges can be recursively approximated, by the Limit Lemma. We can then restrain computations only if and until the approximation tells us that they are defined. Actually, we need to approximate questions of the form $\{e\}^{A\oplus D}(x)$ or $\{e\}^{B\oplus D}(x)$, and A and B are being built. But the Fixed-Point Theorem also allows us to approximate these questions recursively, since we can use A and B in their own definitions. Note that the construction produces only finite restraints, and it is thus a finite injury priority argument.)

In particular, part d) shows that witnesses for the failure of splitting and density combined cannot be found below low r.e. degrees. Harrington has extended this to low$_2$, see Shore and Slaman [1990] for a proof. In the other direction, Shore and Slaman [1993] show that such witnesses can be found below any high r.e. degree.

Exercises X.6.17 Cuppable and cupping degrees. An r.e. degree is called **(non)cuppable** if it is (not) part of a pair of r.e. degrees joining to $\mathbf{0}'$, and **(non) cupping** if (not) every nontrivial r.e. degree below it is part of a pair joining to it. Similarly for r.e. *wtt*-degrees.

a) **Non-Cupping Theorem for $\boldsymbol{\mathcal{R}_{wtt}}$** (Ladner and Sasso [1975]) *There exists a nontrivial noncuppable r.e. wtt-degree*, i.e. $\mathbf{0}'_{wtt}$ is noncupping. (Hint: we want a nonrecursive r.e. set A such that, whenever $A \oplus B \equiv_{wtt} \mathcal{K}$, then $B \equiv_{wtt} \mathcal{K}$. This automatically implies $A <_{wtt} \mathcal{K}$, otherwise it would always be $A \oplus B \equiv_{wtt} \mathcal{K}$, independently of B. The requirements are:

$$\begin{aligned} P_e &: \mathcal{W}_e \text{ infinite } \Rightarrow \mathcal{W}_e \cap A \neq \emptyset \\ N_e &: \mathcal{K} \simeq \{(e)_1\}^{A \oplus \mathcal{W}_{(e)_2}} \text{ with bound } \varphi_{(e)_3} \Rightarrow \mathcal{K} \leq_{wtt} \mathcal{W}_{(e)_2}. \end{aligned}$$

We might think of using the modified length of agreement as in X.6.16.a: as there, this would satisfy N_e, but now it might be $\mathcal{K} \simeq \{(e)_1\}^{A \oplus \mathcal{W}_{(e)_2}}$, e.g. if $\mathcal{W}_{(e)_2} \equiv_{wtt} \mathcal{K}$. Then the restraint might go to infinity, leaving P_e unsatisfied. Note that this also happens with the usual length of agreement function, so some repair must be done anyway. The idea is simply to add permitting to the construction. More precisely, given a length of agreement, we keep the restraint as long as the $\mathcal{K}$-side does not change (to satisfy N_e), but when a new element enters $\mathcal{K}$ below some length of agreement, we consider *it* to be the new length of agreement (to satisfy P_e). Let $l(e,s)$ be the new length of agreement so defined, and $r(e,s) = \max\{u(e,x,s) : x \leq l(e,s)\}$.

Now N_e is satisfied because, if $\mathcal{K} \simeq \{(e)_1\}^{A \oplus \mathcal{W}_{(e)_2}}$ with bound $\varphi_{(e)_3}$, and N_e is not injured after s_0, given x we go to $s > s_0$ such that $l(e,s) > x$ and $\mathcal{W}_{(e)_2}$ has settled on every element up to $f(x) = \max_{y \leq x} \varphi_{(e)_3}(y)$. Then $\mathcal{K}_s(x) = \mathcal{K}(x)$, so that $\mathcal{K} \leq_{wtt} \mathcal{W}_{(e)_2}$ with bound f.

To show that P_e is satisfied, let $I = \{i \leq e : \lim_{s\to\infty} l(i,s) = \infty\}$ and suppose $\mathcal{W}_e \cap A = \emptyset$ and $\mathcal{W}_e$ is infinite. If $i \leq e$ is not in I, then it imposes a finite restraint, so that for almost every $x \in \mathcal{W}_e$ there is a stage such that

$$x \in \mathcal{W}_{e,s} \ \wedge\ x \geq 2e \ \wedge\ x > \max_{i \leq e \,\wedge\, i \notin I} r(i,s) \ \wedge\ x < \min_{i \in I} r(i,s).$$

Let $y \leq x$ be the least such that $x < \min_{i \in I} \max_{z \leq y} u(i,z,s)$. Then $\mathcal{K}_s[y] = \mathcal{K}[y]$, otherwise x would be permitted after s. But then $\mathcal{K}$ is recursive, since $\mathcal{W}_e$ is infinite.)

b) *Every nonzero r.e. wtt-degree is noncupping.* (Ladner and Sasso [1975]). (Hint: change $\mathcal{K}$ into any nonrecursive C in part a) and add permitting to the construction.)

c) *There exists a nonzero noncupping r.e. T-degree.* (Lachlan [1966a]) (Hint: see X.8.18.)

Downey and Jockusch [1987] have proved that *the wtt-degree of a hypersimple set is noncuppable*, thus providing a natural example for the assertion of part a).

Yates and Cooper have proved the **Non-Cupping Theorem for $\mathcal{R}$**, see Miller [1981a] for a proof. As usual, the translation of the proof from the r.e. *wtt*-degrees becomes an infinite injury argument. The novel feature here is that the strategy of part a) may produce $\liminf_{s\to\infty} r(e,s) = \infty$ when $\mathcal{K} \simeq \{(e)_1\}^{A \oplus \mathcal{W}_{(e)_2}}$, since we cannot drop the restraints (otherwise we would not be able to argue that $\mathcal{K} \leq_T \mathcal{W}_{(e)_2}$). Thus this result cannot be treated by the methods of Section 3. Harrington [1978] has proved that, in contrast to part b), *there exists a nonzero cupping r.e. degree*.

The classes of cappable and cuppable r.e. degrees are nontrivial (i.e. neither they nor their complements are empty) for both *wtt*-degrees and *T*-degrees. Harrington [1978] has proved that for Turing degrees they are exhaustive (i.e. every r.e. degree is cappable or cuppable) but not disjoint, see Fejer and Soare [1981] for a proof. For *wtt*-degrees, Ambos-Spies [1985c] and Ambos-Spies, Jockusch, Shore and Soare [1984] have proved that they are nonexhaustive but disjoint.

Exercises X.6.18 Mitotic sets and autoreducible r.e. sets. An r.e. set A is **mitotic** if it can be decomposed in the disjoint union of two r.e. sets B and C of the same degree, i.e. $A \equiv_T B \equiv_T C$. A set A is **autoreducible** (V.5.15) if, for each x, the question 'is x in A?' can be answered recursively in A, without ever asking the oracle about x.

a) *Every r.e. degree contains a mitotic set.* (Hint: consider $A \oplus A$.)

b) *There exists a nonmitotic set.* (Lachlan [1967a]) (Hint: since when B and C are disjoint r.e. sets and $B \cup C = A$, then $B, C \leq_T A$, we can only ensure that the opposite does not hold. The requirements are:

$$R_e : A \simeq \{(e)_1\}^{\mathcal{W}_{(e)_2}} \simeq \{(e)_3\}^{\mathcal{W}_{(e)_4}} \Rightarrow \mathcal{W}_{(e)_2} \cap \mathcal{W}_{(e)_4} \neq \emptyset \vee \mathcal{W}_{(e)_2} \cup \mathcal{W}_{(e)_4} \neq A.$$

To satisfy R_e, choose $z_e \notin A_s$ and wait for $t > s$ such that

$$\begin{gathered} A_t(z_e) \simeq \{(e)_1\}_t^{\mathcal{W}_{(e)_2,t}}(z_e) \simeq \{(e)_3\}_t^{\mathcal{W}_{(e)_4,t}}(z_e) \\ \mathcal{W}_{(e)_2,t} \cap \mathcal{W}_{(e)_4,t} = \emptyset \\ \mathcal{W}_{(e)_2,t} \cup \mathcal{W}_{(e)_4,t} = A_t, \end{gathered}$$

the last condition holding up to the greatest element u used in the computations appearing in the first condition. Spoil this by putting z_e into A and restraining from entering A all elements up to u which are not yet in it. If at the end $\mathcal{W}_{(e)_2} \cap \mathcal{W}_{(e)_4} = \emptyset$ and $\mathcal{W}_{(e)_2} \cup \mathcal{W}_{(e)_4} = A$, then at least one of the two computations $\{(e)_1\}_t^{\mathcal{W}_{(e)_2,t}}(z_e)$ and $\{(e)_3\}_t^{\mathcal{W}_{(e)_4,t}}(z_e)$ has been preserved: indeed, z_e was the only possible element below it u that could enter A, and it could go into only one of $\mathcal{W}_{(e)_2}$ and $\mathcal{W}_{(e)_4}$. But $A_t(z_e)$ has changed, and so either $A \not\simeq \{(e)_1\}^{\mathcal{W}_{(e)_2}}$ or $A \not\simeq \{(e)_3\}^{\mathcal{W}_{(e)_4}}$.)

c) *There exists a complete nonmitotic set.* (Ladner [1973]) (Hint: the construction above gives this automatically, see X.1.5. Or code $\mathcal{K}$ into A directly.)

d) *Below any nonzero r.e. degree there is a nonmitotic degree.* (Ladner [1973]) (Hint: use X.2.7.)

e) *An r.e. set is mitotic if and only if it is autoreducible.* (Ladner [1973]) (Hint: let B and C a mitotic decomposition of A, and $A \simeq \{(e)_1\}^B \simeq \{(e)_2\}^C$. To see if $x \in A$ without using the oracle A on x, approximate the computations $\{(e)_1\}^B(x)$ and $\{(e)_2\}^C(x)$ as follows. If one of the oracles B and C is queried about $y \neq x$, give the right answer recursively in A. If one of the oracles is queried about x, answer that x is not in the set. One of these approximations is the correct computation, since x is in at most one of B and C. Thus, if they output the same value this is the correct value of $A(x)$; if they output different values, then it must be $x \in A$.

Now let A be autoreducible via e, i.e. $A(x)$ is computed by $\{e\}^A(x)$ without using the oracle on x. If $A_s(x) \simeq \{e\}_s^{A_s}(x)$ and $x \in A_{s+1} - A_s$, then $A_{s+1} \not\simeq \{e\}_{s+1}^{A_{s+1}}(x)$, since $\{e\}_s^{A_s}(x) \simeq \{e\}_{s+1}^{A_{s+1}}(x)$ because the oracle does not query x by hypothesis. But $A(x) \simeq \{e\}^A(x)$, and thus there must be $y \neq x$ such that $y \leq u(e, x, s)$, and which enters A at some stage after $s + 1$. In the construction, put e.g. x into B and y into C.)

There are nonzero r.e. degrees whose *r.e.* sets are all mitotic and hence autoreducible (Ladner [1973]), but there is no nonzero r.e. degree whose sets are *all* autoreducible (Jockusch and Paterson [1976]). For more results on this subject see also Cenzer [1977], Ingrassia [1981], Downey and Slaman [1989], Downey and Stob [1993]. In the latter, a number of other notions of splitting are also surveyed.

We conclude our treatment of l.u.b.'s with a brief look at l.u.b.'s of chains of r.e. degrees.

Proposition X.6.19 *There exists a chain of r.e. degrees with no greatest element, and with l.u.b. in the r.e. degrees.*

Proof. By Zorn's Lemma, there exists a maximal chain of r.e. degrees strictly below $\mathbf{0}'$. Such a chain cannot have an upper bound (and, in particular, a greatest element) below $\mathbf{0}'$, otherwise by density it would be extendable, contradicting maximality. Then $\mathbf{0}'$ is its l.u.b. $\square$

While $\mathbf{0}'$ is the l.u.b. in $\boldsymbol{\mathcal{R}}$ of the chain produced above, by V.4.10 it is only a minimal upper bound in $\boldsymbol{\mathcal{D}}$ of such a chain. The next result is the effective analogue of V.4.10.

Proposition X.6.20 (Sacks [1963])

1. *An r.e. chain of r.e. degrees has l.u.b. in the r.e. degrees if and only if it is eventually constant.*
2. *An r.e. set of r.e. degrees has l.u.b. in the r.e. degrees if and only if there is a finite subset of it whose join provides an upper bound for the whole set.*

Proof. Let $\{\mathcal{W}_{f(e)}\}_{e\in\omega}$ be a recursive enumeration of representatives of the r.e. degrees of an r.e. chain which is not eventually constant. Let

$$\langle e, x\rangle \in B \ \Leftrightarrow\ x \in \mathcal{W}_{f(e)}.$$

Given an r.e. nonrecursive set C, by the Thickness Lemma X.3.7 there is a thick r.e. subset A of B such that $C \not\leq_T A$, whenever

$$C \not\leq_T B^{[<e]} = \bigoplus_{n<e} B^{[n]} = \bigoplus_{n<e} \mathcal{W}_{f(n)},$$

in particular if C is an upper bound of the chain (since there is no greatest element). But A is still an upper bound of the chain, so C is not the l.u.b.

Part 2 follows from part 1 as in V.4.10. $\square$

Corollary X.6.21 *Every r.e. chain of r.e. degrees not containing* $\mathbf{0}'$ *is bounded below* $\mathbf{0}'$.

Proof. If the chain has a greatest element, this is less than $\mathbf{0}'$. If there is no greatest element, let $C = \mathcal{K}$ in the proof above. Then A is an incomplete upper bound of the chain. $\Box$

Cooper [1972b] has proved that some r.e. chains of r.e. degrees have r.e. minimal upper bounds (which, by the corollary, cannot be $\mathbf{0}'$). See also Ambos-Spies [1984b]. From X.6.26.b, it follows that some ascending r.e. chains of r.e. degrees have r.e. exact pairs. By extending X.6.8.c, Ambos-Spies [1980] has shown that this is not always the case (and thus *the effective analogue of Spector's Theorem V.4.3 fails for r.e. degrees*). For more information on r.e. upper bounds of sets of r.e. degrees, see Yates [1966], Robinson [1971] and Ambos-Spies [1980].

Since there is a low ascending chain of r.e. degrees without exact pais, *not every Σ_3^0 ideal of $\boldsymbol{\mathcal{R}}$ bounded below $\mathbf{0}'$ has an exact pair*. In contrast to this, Ambos-Spies, Nies and Shore [1992] have proved that *an ideal of $\boldsymbol{\mathcal{R}_{wtt}}$ has an exact pair if and only if it is bounded below $\mathbf{0}'_{wtt}$ and is Σ_3^0*. The conditions are obviously necessary, since for every r.e. set A the index set $\{x : \mathcal{W}_x \leq_{wtt} A\}$ is Σ_3^0 by X.9.14.

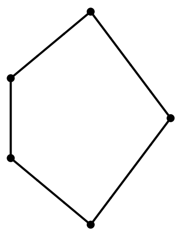
Lattice embeddings $\star$

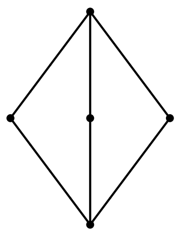
Since every pair of r.e. degrees has an r.e. l.u.b., the existence of minimal pairs shows that a simple distributive lattice (namely, the diamond) is embeddable in the r.e. degrees by *preserving l.u.b.'s and g.l.b.'s*. A complete characterization of the lattices which are embeddable in the r.e. degrees is not known (for partial results, see Ambos-Spies and Lerman [1986], [1989], Lempp and Lerman [1997], Lerman [1998], [199?]; for a survey, see Lerman [1996]). The following results show that such a characterization might be complicated:

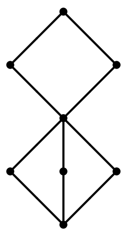
1. *every countable distributive lattice is embeddable* (Thomason [1971], Lachlan, Lerman), by a modification of X.6.5

2. *some nondistributive lattices are embeddable* (Lachlan [1972b]). More precisely, N_5 and M_5 in the picture below are embeddable (see X.6.27.a)

3. *not all finite lattices are embeddable* (Lachlan and Soare [1980]). More precisely, S_8 in the picture below is not embeddable.


N_5


M_5


S_8

The proof of X.6.5 shows that the diamond lattice is actually embeddable in the r.e. degrees by *preserving the least element*, and the same holds of the other embeddings quoted in 1 and 2 above. Lachlan [1980], Soare and Shoenfield have proved that the diamond can be embedded by *preserving the greatest element*, and the same holds of the other embeddings quoted in 1 and 2 above (Ambos-Spies [1980], Ambos-Spies, Lempp and Lerman [1994]).

For the r.e. wtt-degrees, the sublattices that can be embedded are exactly the countable distributive lattices (Stob [1983]). The condition is necessary by X.6.27.c. That it is also sufficient follows from the fact that the embeddability proofs of countable distributive lattices in the r.e. Turing degrees work for the r.e. *wtt*-degrees as well. In particular, all countable distributive lattices can be embedded by preserving the least element or, alternatively, the greatest.

We now show that the diamond cannot be embedded in the r.e. degrees by *preserving both least and greatest element*, although some lattices can be so embedded (Ambos-Spies [1980], Ambos-Spies, Lempp and Lerman [1994a]).

Proposition X.6.22 (Lachlan [1966b], Jockusch) *If two r.e. degrees are part of minimal pairs, they cannot join to* $\mathbf{0}'$.

Proof. We prove the contrapositive, i.e. that given r.e. sets A and B joining to $\mathcal{K}$, there are no r.e. sets $\widehat{A}$ and $\widehat{B}$ such that A and B form minimal pairs with them, respectively.

Let A, $\widehat{A}$, B and $\widehat{B}$ be r.e. nonrecursive sets such that $A \oplus B \equiv_T \mathcal{K}$, and let $\{\hat{a}_s\}_{s\in\omega}$ and $\{\hat{b}_s\}_{s\in\omega}$ be enumerations of $\widehat{A}$ and $\widehat{B}$, respectively.

- *primary strategy*
  We try to avoid A and $\widehat{A}$ forming a minimal pair, by building $E \leq_T A, \widehat{A}$ such that E is not recursive. The requirements are

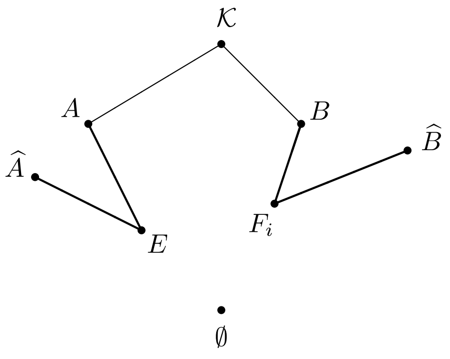
$$R_i \ : \ \mathcal{W}_i \text{ infinite } \Rightarrow \ E \cap \mathcal{W}_i \neq \emptyset.$$

- *secondary strategy*
  If we fail to satisfy R_i for some i, we try to avoid B and $\widehat{B}$ forming a

minimal pair, by building $F_i \leq_T B, \widehat{B}$ such that F_i is not recursive. The requirements are

$$R_{i,j} \; : \; \mathcal{W}_j \text{ infinite } \Rightarrow \; F_i \cap \mathcal{W}_j \neq \emptyset.$$

The situation can be depicted as follows:



It is equivalent to consider only the following requirements:

$$P_{i,j} \; : \; \mathcal{W}_i, \mathcal{W}_j \text{ infinite } \Rightarrow \; E \cap \mathcal{W}_i \neq \emptyset \text{ or } F_i \cap \mathcal{W}_j \neq \emptyset.$$

If they are all satisfied, then either R_i is satisfied for all i, or $R_{i,j}$ is satisfied for some i and all j.

The problem is clearly how to push E below A and $\widehat{A}$, or F_i below B and $\widehat{B}$. One side can easily be ensured by permitting, and for the other we use delayed permitting (X.6.10). The idea is that there is a way of forcing a change at some point in one of A and B, by building an auxiliary r.e. set D. Since $A \oplus B$ is complete, $D \simeq \{e\}^{A \oplus B}$ for some e, which we may suppose to know in advance by the Fixed-Point Theorem. If at a certain stage we have

$$D_s(z) \simeq \{e\}_s^{A_s \oplus B_s}(z) \simeq 0,$$

by putting z into D we force a change into $A \oplus B$ (below the use function), and so into one of A and B.

We thus attack a requirement $P_{i,j}$ only when we have elements ready to satisfy both sides. Since only one element is needed to satisfy $P_{i,j}$, we can devote $\langle i, j \rangle$ to force the change into A or B for the sake of $P_{i,j}$. We thus wait for stages:

- t such that $\{e\}_t^{A_t \oplus B_t}(\langle i, j \rangle) \simeq 0$ with use $u(e,t)$; and, for some y,

$$y \in \mathcal{W}_{j,t} \; \wedge \; y \geq 2j \; \wedge y > u(e,t) \; \wedge \; y > \hat{b}_t,$$

(the last clause showing that y is permitted by $\widehat{B}$ at t)

- $s > t$ such that $\{e\}_s^{A_s \oplus B_s}(\langle i, j\rangle) \simeq 0$ with use $u(e, s)$; for some x,

  $$x \in \mathcal{W}_{i,s} \ \wedge \ x \geq 2i \ \wedge x > u(e, s) \ \wedge \ x > \hat{a}_s,$$

  (the last clause showing that x is permitted by $\widehat{A}$ at s); and

  $$A_s \oplus B_s \text{ and } A_t \oplus B_t \text{ look the same up to } u(e, t).$$

Then put $\langle i, j\rangle$ in D at stage $s+1$. This forces a change below $u(e, s)$, since now $\{e\}^{A\oplus B}(\langle i, j\rangle) \simeq 1$. Enumerate A and B to see where the change occurs, and hence which of the two following cases happen: either x is permitted by both A and $\widehat{A}$, or y is permitted by both B and $\widehat{B}$. In the first case put x into E, in the second put y into F_i. This is the construction at step $s+1$, if $P_{i,j}$ is the least condition (in any effective ordering of them) that is not yet satisfied, and that can be satisfied as above.

By permitting, we have $E \leq_T A, \widehat{A}$ and $F_i \leq_T B$. The only delayed permitting occurs when y goes into F_i, since y was permitted by $\widehat{B}$ only before stage s. If E is nonrecursive, then we have satisfied the requirements R_i, so suppose $E = \overline{\mathcal{W}}_i$ for some i. We want to show that F_i is nonrecursive and $F_i \leq_T \widehat{B}$:

- *F_i is nonrecursive*
  Each $P_{i,j}$ is satisfied (since $\widehat{A}$ and $\widehat{B}$ are nonrecursive, so separately they permit sufficiently many times), and it must be so because of the F_i-side, otherwise $E \cap \mathcal{W}_i \neq \emptyset$.

- *$F_i \leq_T \widehat{B}$*
  Given y, search (recursively in $\widehat{B}$) for the stages in which it is permitted by $\widehat{B}$, of which there are only finitely many. For each such t, find $s > t$ such that either $A_s \oplus B_s$ and $A_t \oplus B_t$ do not agree up to $u(e, s)$ (in which case we do not satisfy $P_{i,j}$ using y, so $y \notin F_i$), or y goes into F_i (since $P_{i,j}$ is satisfied only on the F_i-side). □

Corollary X.6.23 Non-Diamond Theorem for $\mathcal{R}$ (Lachlan [1966b]) *No minimal pair of r.e. degrees can join to* $\mathbf{0}'$.

Since permitting works for r.e. *wtt*-degrees (p. I.338), the same proof gives the **Non-Diamond Theorem for $\mathcal{R}_{wtt}$**.

Unlike for the degrees below $\mathbf{0}'$ (see XI.3.1), *there is no pair of r.e. degrees such that every nontrivial r.e. degree is incomparable with one of them.* Indeed, given any two nontrivial degrees, either they do not form a minimal pair, and thus there is a nontrivial degree below both of them, or they do not join to $\mathbf{0}'$, and thus there is a nontrivial degree above both of them. Harrington [1978] has shown that there are *three* degrees such that every nontrivial r.e. degree is incomparable with one of them.

Corollary X.6.24 Non-Capping Theorem for $\mathcal{R}$ (Yates [1966]) *There are nontrivial r.e. degrees that are not part of a minimal pair.*

Proof. By the Sacks Splitting Theorem, there are nontrivial r.e. degrees joining to $\mathbf{0}'$. By X.6.22, they cannot both be part of a minimal pair. □

The **Non-Capping Theorem for $\mathcal{R}_{wtt}$** can be proved either by the same proof as above, or by an analogue of the direct proof in X.6.9.a.

Exercises X.6.25 a) *The use of the Fixed-Point Theorem in the proof of X.6.22 can be eliminated.* (Lerman [1974]) (Hint: for any fixed e, the construction of X.6.22 uniformly defines an r.e. set D_e, where one now puts $\langle e, \langle i, j\rangle\rangle$, rather than $\langle i, j\rangle$, into D_e, and with the possibility that $\{e\}^{A\oplus B}(\langle e, \langle i, j\rangle\rangle)$ does not converge in general. By uniformity, $D = \bigoplus_{e\in\omega} D_e$ is r.e., and so $D \leq_T A \oplus B$. Then there is e_0 such that $D \simeq \{e_0\}^{A\oplus B}$, and for this e_0 the construction succeeds.)

b) *There is no uniform proof of X.6.23*, i.e. there is no recursive function f such that if $A = \mathcal{W}_a$, $B = \mathcal{W}_b$ and $\mathcal{K} \simeq \{e\}^{A\oplus B}$, then $E \simeq \{f(a,b,e)\}^A \simeq \{f(a,b,e)\}^B$ is not recursive. (Lachlan [1966b]) (Hint: for any single minimal pair requirement

$$N_e \ : \ \{e\}^A \simeq \{e\}^B \simeq C \ \Rightarrow \ C \text{ recursive}$$

we can build r.e. sets A and B satisfying N_e, and such that $\mathcal{K} \leq_T A \oplus B$. When x shows up in $\mathcal{K}$, enumerate some $y \leq 3x$ in A or B. This gives $\mathcal{K} \leq_T A \oplus B$, and allows us to make A and B simple. After proving this, use the Fixed-Point Theorem to prove the claim.)

For a variant of the proof of X.6.23 see Ambos-Spies [1984b].

Exercises X.6.26 a) *No pair of incomparable low r.e. degrees joining to $\mathbf{0}'$ has g.l.b.* (Lachlan [1966b]) (Hint: if the g.l.b. existed, it would be a low degree $\boldsymbol{a}$. But the Non-Diamond Theorem relativized to $\boldsymbol{a}$ shows that no pair of r.e. degrees with g.l.b. $\boldsymbol{a}$ can join to $\boldsymbol{a}' = \mathbf{0}'$.)

b) *Every pair of incomparable low r.e. degrees joining to $\mathbf{0}'$ is the exact pair of an ascending r.e. chain of r.e. degrees.* (Ambos-Spies [1984b]) (Hint: the index sets of $\{x : \mathcal{W}_x \leq_T A\}$ and $\{x : \mathcal{W}_x \leq_T B\}$ are Σ^0_3 by lowness, see X.9.15, and hence so is $\{x : \mathcal{W}_x \leq_T A, B\}$. Then this family is r.e. by II.5.25. If f enumerates it, then $\{\bigoplus_{n\leq x} \mathcal{W}_{f(n)}\}_{x\in\omega}$ is uniformly r.e., has no greatest element by part a), and A and B are an exact pair for it.)

Exercises X.6.27 Distributivity. a) *$\mathcal{R}$ is not distributive.* (Lachlan [1972b]) (Hint: we embed the pentagon lattice, which is nondistributive, in the r.e. degrees. We build A, B and C r.e. such that A and C form a minimal pair, and

$$A <_T A \oplus B <_T A \oplus B \oplus C \equiv_T A \oplus C.$$

The requirements are:

$$\begin{array}{lcl} R_{3e} & : & A \not\simeq \{e\} \\ R_{3e+1} & : & C \not\simeq \{e\} \\ R_{3e+2} & : & B \not\simeq \{e\}^A \\ N_e & : & \{e\}^{A\oplus B} \simeq \{e\}^C \simeq D \ \Rightarrow\ D \text{ recursive.} \end{array}$$

R_{3e} and R_{3e+1} make A and C nonrecursive, R_{3e+2} makes A strictly lower than $A\oplus B$, and N_e makes $A \oplus B$ and C a minimal pair. Moreover, we also want $B \leq_T A \oplus C$, to have $A \oplus B \oplus C \equiv_T A \oplus C$.

N_e is satisfied as in the minimal pair construction, and R_{3e} and R_{3e+1} are satisfied in the usual way. For R_{3e+2}, we must work in such a way as to ensure that $B \leq_T A\oplus C$. Thus, when we decide to put x into B, then one of A and C has to know it. We cannot simply put x into A, since this might change the computation $\{e\}_s^{A_s}(x) \simeq 0$, which is why we want to put x in B. Similarly, we cannot simply put x into C, since this would ruin the minimal pair strategy. We can only put elements in one side at the same time, and thus not simultaneously into $A \oplus B$ and C.

The solution is similar to that of X.6.8.c. Namely, when x is appointed as a follower, we define a_x^s as a trace associated with x and we want a_x to go in C when x goes into B. When $\{e\}_s^{A_s}(x) \simeq 0$, then put a_x in C as soon as N_e allows it, and appoint a new trace b_x greater than the use of $\{e\}_s^{A_s}(x)$. Wait until N_e allows, and then put b_x into A (this does not destroy the computation, since b_x was appropriately chosen), and x into B (this puts a second element in, but on the same side).

To prove $B \leq_T A \oplus C$, first see if x is a follower. If so, we also know a_x, which is appointed at the same stage in which x was appointed as a follower. See if $a_x \in C$. If not, $x \notin B$. Otherwise, find the stage at which a_x goes into C, which is also the stage at which b_x was defined. See if $b_x \in A$. If so, $x \in B$.)

b) $\boldsymbol{\mathcal{D}}$ *is not distributive.* (Lerman [1969], Shoenfield) (Hint: by X.6.12.a, the g.l.b. of two r.e. degrees is the same in the r.e. degrees and in the degrees.)

c) $\boldsymbol{\mathcal{R}_{wtt}}$ *is distributive.* (Lachlan [1972b]) (Hint: if $A \leq_{wtt} B \oplus C$ build $\widehat{B} \leq_{wtt} B$ and $\widehat{C} \leq_{wtt} C$ such that $A = \widehat{B} \cup \widehat{C}$ and $\widehat{B} \cap \widehat{C} = \emptyset$. Let $A \simeq \{e\}^{B\oplus C}$ with use φ_i. If $x \in A_{s+1} - A_s$, let $t \leq s$ be the greatest stage such that $\{e\}_t^{B_t\oplus C_t}(x)$ converges. If t does not exist, or the value of the computation is not 0, put x in $\widehat{B}$. Otherwise, the value is 0. Since $x \in A$, i.e. $\{e\}^{B\oplus C}(x) \simeq 1$, there is $t' > t$ such that some integer $y \leq \varphi_i(x)$ enters $B \oplus C$ at stage t'. Let t' and y be minimal. If y is even (i.e. something enters B at stage t'), put x into $\widehat{B}$, otherwise put x into $\widehat{C}$. Since we may suppose that $x \in A$ whenever $\{e\}_s^{B_s\oplus C_s}(x) \simeq 1$, we have for example $\widehat{B} \leq_{wtt} B$ because $x \in \widehat{B}$ if and only if x is in $\widehat{B}$ at the smallest stage s such that $\{e\}_s^{B_s\oplus C_s}(x)$ converges, and B has settled on all elements up to $\varphi_i(x)$.)

That $\boldsymbol{\mathcal{D}_{wtt}}$ *is not distributive* follows from the fact that one can embed the pentagon as an initial segment (see p. I.529 and I.589).

Definability from parameters

We now look at global properties of $\mathcal{R}$, following the path set up in Section V.7. The first step is to obtain definability results from parameters, and the next one provides a weak analogue of V.7.1.

Proposition X.6.28 (Harrington and Shelah [1982], Nies, Shore and Slaman [1998]) *There is a uniformly low antichain definable in $\mathcal{R}$ from finitely many low parameters.*

Proof. We want to construct a uniformly low antichain $\mathcal{C} = \{\boldsymbol{c}_n\}_{n\in\omega}$ of r.e. degrees, together with low r.e. parameters coding it. Since in the r.e. degrees it is complicated to deal with g.l.b.'s, in place of the property used in V.7.1 we use a simpler property, involving only l.u.b.'s. Namely, given two r.e. degrees $\boldsymbol{a}$ and $\boldsymbol{b}$, we look at the r.e. degrees $\boldsymbol{x}$ satisfying the property P defined as follows:

$$P(\boldsymbol{x}) \Leftrightarrow \boldsymbol{b} \leq \boldsymbol{x} \cup \boldsymbol{a}.$$

If we make sure, as in V.7.1, that the elements of $\mathcal{C}$ are the minimal solutions of P below a third r.e. degree $\boldsymbol{c}$ bounding $\mathcal{C}$, then

$$\boldsymbol{x} \in \mathcal{C} \Leftrightarrow \boldsymbol{x} \leq \boldsymbol{c} \wedge P(\boldsymbol{x}) \wedge \neg(\exists \boldsymbol{z})(\boldsymbol{z} < \boldsymbol{x} \wedge P(\boldsymbol{z})).$$

We will thus define $\mathcal{C}$ using three parameters $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$, of which $\boldsymbol{c}$ is an r.e. degree uniformly bounding every $\boldsymbol{c}_n$.

We construct r.e. sets C_n, A and B, and let $C = \oplus_{n\in\omega} C_n$. The conditions on C_n, A and B are the following:

1. $\{\boldsymbol{c}_n\}_{n\in\omega}$ *is an antichain*
   For this we need, for any $n \neq m$,
   $$\boldsymbol{c_n} \not\leq \boldsymbol{c_m},$$
   which is ensured by the requirements
   $$Z_{e,n,m} \quad : \quad C_n \not\simeq \{e\}^{C_m}.$$
   $Z_{e,n,m}$ is satisfied by direct diagonalization, i.e. by appropriately choosing a witness z_e, that goes into C_n if $\{e\}^{C_m}(z_e) \simeq 0$.

2. $\boldsymbol{c}_n$ *is a solution of* P
   For this we need
   $$\boldsymbol{b} \leq \boldsymbol{c_n} \cup \boldsymbol{a},$$
   which is ensured by the requirements
   $$P_n \quad : \quad B \leq_T C_n \oplus A.$$

P_n is satisfied by standard coding, i.e. by associating to every number y a marker $\gamma_y > y$ that goes into A when y goes into B, and is redefined to a bigger value if C_n changes below its current value.

3. ***$\boldsymbol{c}_n$ is a minimal solution of P***
   For this we need, for any $\boldsymbol{x} \leq \boldsymbol{c}$,

$$\boldsymbol{b} \leq \boldsymbol{x} \cup \boldsymbol{a} \Rightarrow \boldsymbol{c_n} \leq \boldsymbol{x}, \text{ for some } n.$$

   This means that, for any r.e. set $X \leq C$,

$$B \leq_T X \oplus A \Rightarrow C_n \leq_T X, \text{ for some } n,$$

   and is ensured by the requirements

$$N_{p,i,j} \quad : \quad \mathcal{W}_p \simeq \{i\}^C \wedge B \simeq \{j\}^{\mathcal{W}_p \oplus A} \Rightarrow C_n \leq_T \mathcal{W}_p, \text{ for some } n.$$

   $N_{p,i,j}$ is satisfied in a way reminiscent of the minimal pair construction (X.6.5), in particular using maximal stages, but with a dual strategy. Instead of proceeding indirectly, by trying to falsify the premise and arguing that if we fail, then we can satisfy the conclusion, we proceed directly, by trying to satisfy the conclusion and arguing that if we fail, then we can falsify the premise. In practice, instead of trying to diagonalize against $B \simeq \{j\}^{\mathcal{W}_p \oplus A}$ whenever possible, we diagonalize only if we have an apparent failure by $\mathcal{W}_p$ to change whenever a change in C_n occurs.

4. *the antichain and the parameters are uniformly low*
   For this we need

$$C \oplus A \oplus B \text{ low},$$

   which is ensured (see X.3.3) by the requirements:

$$Q_{e,x} \quad : \quad (\exists_\infty s)(\{e\}_s^{C_s \oplus A_s \oplus B_s}(x)\downarrow) \Rightarrow \{e\}^{C \oplus A \oplus B}(x)\downarrow$$

   $Q_{e,x}$ is satisfied as in the construction of low degrees (X.3.3), i.e. by restraining the computations of $\{e\}^{C \oplus A \oplus B}$ as soon as they converge.

We now have to discuss how to combine the first three kinds of strategies (the fourth one is not problematic, since it only imposes additional finitary restraints). We first look at the simpler *construction with only one* C_n, say C_0, in which case the first three kinds of requirements become:

$$\begin{array}{lcl} Z_e & : & C_0 \not\simeq \{e\} \\ P & : & B \leq_T C_0 \oplus A \\ N_{p,i,j} & : & \mathcal{W}_p \simeq \{i\}^{C_0} \wedge B \simeq \{j\}^{\mathcal{W}_p \oplus A} \Rightarrow C_0 \leq_T \mathcal{W}_p. \end{array}$$

For each e, we pick a witness z_e for the diagonalization of Z_e. While waiting for $\{e\}(z_e)$ to converge to 0, we consider the finitely many requirements $N_{p,i,j}$ of priority higher than Z_e. For each of them we see if there is some $y > z_e$ not yet associated to any number, such that $\{j\}^{\mathcal{W}_p \oplus A}(y) \simeq 0$. If so, we associate such a y to z_e, restrain A below the use of the computation and later change z_e if the restraint is violated.[5]

If $\{e\}(z_e)$ converges to 0, we put z_e into C_0 for the sake of Z_e. Since this produces a change in C_0 below γ_y, because $\gamma_y > y > z_e$, for the sake of P we redefine γ_y as the current stage (in particular, γ_y is now bigger than all uses of currently converging computations). Since a change in C_0 might have produced a change in $\mathcal{W}_p$, we see if the value of the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$ is still 0. If so, we put y into B for the sake of $N_{p,i,j}$, and γ_y into A for the sake of P. Notice that this does not interfere with the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$, because γ_y is now bigger than its use.

After we put z_e into C_0 for the sake of Z_e, we still want to preserve A below the use of the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$ until we see a new maximal stage for $N_{p,i,j}$, since only then we can see if $\mathcal{W}_p$ has changed. This produces an additional restraint on A, which is dropped at maximal stages.

We now have to argue that the requirements are satisfied.

- *Z_e is satisfied*

  Since z_e is put into C_0 if and only if $\{e\}(z_e) \simeq 0$, it witnesses that $C_0 \not\simeq \{e\}$.

- *P is satisfied*

  We want to prove $B \leq_T C_0 \oplus A$, i.e. to decide recursively in $C_0 \oplus A$ whether $y \in B$, for every y.

  We can set up the construction in such a way that we define γ_y for the first time at stage y. Since a new definition requires a change in C_0 below the current value of γ_y, and γ_y changes only if y is associated to some z_e, i.e. at most once, we can compute its final value recursively in C_0. Then

  $$y \in B \Leftrightarrow \gamma_y \in A.$$

  Thus we can determine whether y is in B, recursively in $C_0 \oplus A$.

- *$N_{p,i,j}$ is satisfied*

[5] We are omitting the combinatorial problem of choosing potential witnesses z_e and associated elements y in a way that makes the following argument work. In a formal proof, we would need to argue that there is a system of tying potential witnesses and associated elements that either makes the functionals $\{j\}^{\mathcal{W}_p \oplus A}$ total and correct or eventually makes diagonalization possible.

If $\mathcal{W}_p \simeq \{i\}^{C_0}$ and $B \simeq \{j\}^{\mathcal{W}_p \oplus A}$, we want to prove $C_0 \leq_T \mathcal{W}_p$, i.e. to decide recursively in $\mathcal{W}_p$ whether $z \in C_0$, for every z.

We can set up the construction in such a way that at any stage only witnesses greater than that stage are chosen, so that a number z can become a witness for some requirement Z_e, i.e. $z = z_e$, only at stages smaller than z. Given z, at stage z we thus know whether $z = z_e$ for some e. If not, then $z \notin C_0$. Otherwise, we look for the first stage at which some y is associated to z_e, which exists because $B \simeq \{j\}^{\mathcal{W}_p \oplus A}$ by hypothesis. In particular, $\{j\}^{\mathcal{W}_p \oplus A}(y) \simeq 0$.

The crucial observation is that if z ever goes into C_0, then this produces a change in the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$. Otherwise, the construction would put y into B and make $B \not\simeq \{j\}^{\mathcal{W}_p \oplus A}$, contradicting the hypothesis. Moreover, the change in the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$ must occur on $\mathcal{W}_p$, since A is restrained by Z_e.

To decide whether z ever goes into C_0, it is thus enough to see if it has gone in by the first stage after which either the restraint imposed by Z_e on A is injured, in which case z_e is redefined and z ceases to be a witness, or $\mathcal{W}_p$ no longer changes below the use of the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$. This later stage exists by the lowness requirements and it can be found recursively in $\mathcal{W}_p$. In both cases, z is in C_0 if and only if it is already in by that stage.

We now look at the *construction with two C_n's*, say C_0 and C_1, in which case the first three kinds of requirements become:

$$\begin{array}{lcll} Z_{e,n} & : & C_n \not\simeq \{e\}^{C_{1-n}} & (n = 0, 1) \\ P_n & : & B \leq_T C_n \oplus A & (n = 0, 1) \\ N_{p,i,j} & : & \mathcal{W}_p \simeq \{i\}^{C_0 \oplus C_1} \wedge B \simeq \{j\}^{\mathcal{W}_p \oplus A} \quad \Rightarrow & C_0 \leq_T \mathcal{W}_p \text{ or } C_1 \leq_T \mathcal{W}_p. \end{array}$$

For each e_0 and e_1, we pick witnesses z_{e_0} and z_{e_1} for the diagonalization of $Z_{e_0,0}$ and $Z_{e_1,1}$. We consider only the case in which $Z_{e_0,0}$ has higher priority than $Z_{e_1,1}$, since the opposite case is symmetric. While waiting for $\{e_0\}^{C_1}(z_{e_0})$ to converge to 0, we consider the finitely many requirements $N_{p,i,j}$ of priority higher than $Z_{e_0,0}$. For each of them we see if there is some $y > z_{e_0}$ not yet associated to any number, such that $\{j\}^{\mathcal{W}_p \oplus A}(y) \simeq 0$. If so, we associate such a y to z_{e_0} (not yet to z_{e_1}), and restrain A below the use of the computation.

If $\{e_0\}^{C_1}(z_{e_0})$ converges to 0, we put z_{e_0} into C_0 for the sake of $Z_{e_0,0}$, and redefine γ_y^0 for the sake of P_0 (in particular, γ_y^0 is bigger than all uses of currently converging computations). If this does not produce a change in the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$, we now associate y to z_{e_1}, too.

If $\{e_1\}^{C_0}(z_{e_1})$ converges to 0, we put z_{e_1} into C_1 for the sake of $Z_{e_1,1}$, and redefine γ_y^1 for the sake of P_1. If this does not produce a change in the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$, we put y into B for the sake of $N_{p,i,j}$, and both γ_y^0 and γ_y^1 into A for the sake of P_0 and P_1.

In other words, the requirement $Z_{e_0,0}$ looks for y's as before, while the requirement $Z_{e_1,1}$ looks instead for y's that have already gone through the process of having been chosen by a $Z_{e_0,0}$.

We now have to argue that the requirements are satisfied.

- *$Z_{e,n}$ is satisfied*

  As in the proof with only one C_n.

- *P_n is satisfied*

  As in the proof with only one C_n, we reduce B to $C_0 \oplus A$ by using the fact that if y is associated to z_{e_0}, then

  $$y \in B \Leftrightarrow \gamma_y^0 \in A.$$

  Similarly, we reduce B to $C_1 \oplus A$ by using the fact that if y is associated to z_{e_1}, then

  $$y \in B \Leftrightarrow \gamma_y^1 \in A.$$

- *$N_{p,i,j}$ is satisfied*

  If $\mathcal{W}_p \simeq \{i\}^{C_0 \oplus C_1}$ and $B \simeq \{j\}^{\mathcal{W}_p \oplus A}$, we want to prove that $C_0 \leq_T \mathcal{W}_p$ or $C_1 \leq_T \mathcal{W}_p$.

  We can no longer claim, as in the proof for only one C_n, that if z_{e_0} ever goes into C_0 and y is associated to it, then this produces a change in the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$. But we can still claim that either this holds or, if z_e^1 ever goes into C_1, then this produces a change in the computation. We must then proceed by cases.

  Suppose that, for almost every e_0, if z_{e_0} ever goes into C_0 and y is associated to z_{e_0}, then this produces a change in the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$. In this case we can proceed as in the proof for only one C_n and deduce that $C_0 \leq_T \mathcal{W}_p$.

  Otherwise, it is then the case that infinitely often a y that was associated to z_{e_0} becomes also associated to z_{e_1}. We can now claim that if z_{e_1} ever goes into C_1 and y is associated to z_{e_1}, then this produces a change in the computation $\{j\}^{\mathcal{W}_p \oplus A}(y)$. Once again, we can thus proceed as in the proof for only one C_n and deduce that $C_1 \leq_T \mathcal{W}_p$.

The *construction with infinitely many C_n's* is similar to that with only two, because each $N_{p,i,j}$ has to take care only of the finitely many C_n's introduced by requirements of higher priority. One can thus proceed by induction and show that if the hypothesis of $N_{p,i,j}$ is satisfied, then one of these C_n's is reducible to $\mathcal{W}_p$. $\Box$

While V.7.1 was universal, showing how to define in $\boldsymbol{\mathcal{D}}$ from parameters *every* countable antichain, X.6.28 is only existential, and shows how to define in $\boldsymbol{\mathcal{R}}$ from parameters *some* countable antichain. For the applications, one needs greater flexibility, and a small variation on the construction immediately provides a weak analogue of V.7.2.

Proposition X.6.29 Definability from parameters (Harrington and Shelah [1982], Nies, Shore and Slaman [1998]) *Every recursive partial ordering is definable in $\boldsymbol{\mathcal{R}}$ from finitely many low parameters, in a uniform way.*

Proof. We use the methods of X.6.28 to code the integers by an antichain $\{\boldsymbol{c}_n\}_{n\in\omega}$ definable from parameters $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$, and introduce an additional parameter $\boldsymbol{d}$ satisfying the additional condition:

5. *$\boldsymbol{d}$ codes the partial ordering R*
   This can be achieved by letting
   $$R(n,m) \;\Leftrightarrow\; \boldsymbol{c_n} \leq \boldsymbol{d} \cup \boldsymbol{c_m},$$
   and hence by constructing an additional r.e. set D such that, for any n and m,
   $$R(n,m) \;\Leftrightarrow\; C_n \leq_T D \oplus C_m,$$
   which is ensured by the requirements
   $$S_{n,m} \quad : \quad R(n,m) \;\Rightarrow\; C_n \leq_T D \oplus C_m$$
   and
   $$T_{e,n,m} \quad : \quad \neg R(n,m) \;\Rightarrow\; C_n \not\simeq \{e\}^{D\oplus C_m}.$$

Obviously, condition 4 in the proof of X.6.28 is now modified to include D in the uniform lowness requirement.

$T_{e,n,m}$ is satisfied in the same way as $Z_{e,n,m}$, by choosing a witness t_e for the diagonalization, that goes into C_n if $\{e\}^{D\oplus C_m}(t_e) \simeq 0$. Moreover, $T_{e,n,m}$ imposes restraints in the same way as $Z_{e,n,m}$ does.

$S_{n,m}$ is satisfied by direct coding, e.g. by requiring that if an element goes into C_n, then it also goes into D or C_m. There are only two reasons why an element may ever want to go into C_n: either it is a z_e trying to satisfy a $Z_{e,n,m'}$, or it is a t_e trying to satisfy a $T_{e,n,m'}$. In the first case nothing prevents the possibility that $m = m'$, and we thus put z_e into D to avoid conflicts. In the second case, instead, it is safe to put t_e into C_m, because certainly $m \neq m'$. Indeed, $S_{n,m}$ is considered only if $R(n,m)$, while $T_{e,n,m'}$ only if $\neg R(n,m')$. □

An additional variation of the proof of X.6.28 provides a more substantial improvement, and shows that the elements of the definable antichain can satisfy various restrictions, such as (not) being below given low r.e. degrees.

Proposition X.6.30 (Nies, Shore and Slaman [1998]) *For any k, given $k+1$ nonzero low r.e. degrees*

$$\boldsymbol{e}_0, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_{k-1} \qquad \textit{and} \qquad \boldsymbol{e},$$

there is a uniformly low antichain $\{\boldsymbol{c}_n\}_{n\in\omega}$ definable in $\boldsymbol{\mathcal{R}}$ from finitely many low parameters, and such that

$$\begin{array}{ll} \boldsymbol{c}_n \leq \boldsymbol{e}_n & \textit{if } n < k \\ \boldsymbol{c}_n \not\leq \boldsymbol{e} & \textit{otherwise} \end{array}$$

and, for $n, m < k$,

$$\boldsymbol{e}_n \not\leq \boldsymbol{e}_m \Rightarrow \boldsymbol{c}_n \not\leq \boldsymbol{e}_m.$$

Proof. We use the methods of X.6.28 to define an antichain $\{\boldsymbol{c}_n\}_{n\in\omega}$ from parameters $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$, satisfying conditions 1–4, as well as the following additional ones, where $E_n \in \boldsymbol{e}_n$ and $E \in \boldsymbol{e}$:

6. *comparability conditions*
   To ensure, for $n < k$,
   $$\boldsymbol{c}_n \leq \boldsymbol{e}_n$$
   we satisfy
   $$U_n \quad : \quad C_n \leq_T E_n.$$

7. *incomparability conditions*
   To ensure, for $n \geq k$,
   $$\boldsymbol{c}_n \not\leq \boldsymbol{e}$$
   we satisfy
   $$V_{e,n} \quad : \quad C_n \not\simeq \{e\}^E.$$
   To ensure, for $n, m < k$,
   $$\boldsymbol{e}_n \not\leq \boldsymbol{e}_m \Rightarrow \boldsymbol{c}_n \not\leq \boldsymbol{e}_m,$$
   we satisfy, for every n and m such that $E_n \not\leq_T E_m$,
   $$V_{e,n,m} \quad : \quad C_n \not\simeq \{e\}^{E_m}.$$

U_n is satisfied by permitting (III.3.18), that is by asking that numbers go into C_n only when they are permitted by E_n, i.e. only at stages when a smaller element is generated in E_n. This produces the following modification in the strategy for $Z_{e,n,m}$. Instead of defining a single witness z_e, we define a recursive list of them, each of which has to go through the whole process associated with the higher negative priority requirements. Then we put in C_n the first witness for which $\{e\}^{C_m}$ converges to 0 *and* is permitted by E_n. The modified strategy succeeds because E_n is not recursive, and hence one of the suitable witnesses is eventually permitted to go in C_n.

$V_{e,n}$ is satisfied as $Z_{e,n,m}$, by choosing a witness v_e for the diagonalization, which is put into C_n if $\{e\}^E(v_e) \simeq 0$. Moreover, $V_{e,n}$ imposes restraints in the same way as $Z_{e,n,m}$ does. A problem arises from the fact that $V_{e,n}$ could be satisfied because $\{e\}^E$ is undefined on the least disagreement, thus pushing the restraint function to infinity. But since E is low, the question of whether $\{e\}^E(x)$ converges can be recursively approximated, by the Limit Lemma. We can then restrain computations only if and until when the approximations tell us that they are defined, thus producing only finite restraints.

$V_{e,n,m}$ is satisfied as $V_{e,n}$, using the fact that E_m is low, with the additional condition that, because of U_n, a witness can be put into C_n only when it is permitted by E_n. Now the permitting strategy succeeds because E_n is not recursive in E_m. $\Box$

Notice that the last condition in the statement of X.6.30 ensures that if $\{\boldsymbol{e}_i\}_{i<k}$ is an antichain, then $\boldsymbol{c_n}$ is the only r.e. degree in the antichain $\{\boldsymbol{c}_i\}_{i<k}$ that is below $\boldsymbol{e_n}$ (nothing can be said in general about the relationship between $\boldsymbol{e_n}$ and $\boldsymbol{c_i}$, when $i \geq k$).

Although far from being the best possible ones, the previous definability results are sufficiently flexible to allow the coding of standard models of arithmetic in $\boldsymbol{\mathcal{R}}$, as we now see.

The complexity of the theory of r.e. degrees

The local version of Simpson's Theorem (V.7.3) reads as follows.

Theorem X.6.31 Harrington and Slaman's Theorem (Harrington and Slaman) *The first-order theory of $\boldsymbol{\mathcal{R}}$ has the same degree (and actually the same isomorphism type) as the theory of First-Order Arithmetic.*

Proof. We prove that the two-theories have the same m-degree by interpreting each in the other, thus providing faithful translations that will preserve theorems. Since the translations will actually be one-one, the theories will have the same 1-degree, and hence will be recursively isomorphic by III.7.13.

One direction is clear, since every formula about the ordering of the r.e. degrees can be interpreted in a natural way as a formula about r.e. sets of integers, and hence about their r.e. indices. Thus the theory of r.e. degrees is interpretable in First-Order Arithmetic.

For the converse, we want to show that First-Order Arithmetic is interpretable in $\boldsymbol{\mathcal{R}}$. The proof elaborates the ideas used in X.5.27, and is devoted to showing on the one hand that we can express in $\boldsymbol{\mathcal{R}}$ the fact that given parameters code a standard model of arithmetic, and on the other hand that such parameters exist in the r.e. degrees.

Having done this, then a sentence φ of First-Order Arithmetic is true if and only if the sentence φ^* of $\boldsymbol{\mathcal{R}}$ is true, where φ^* states that for any parameters coding a standard model of arithmetic in the r.e. degrees, the translation of φ holds. Since φ^* can be effectively obtained from φ, we thus have an m-reduction of First-Order Arithmetic to the theory of $\boldsymbol{\mathcal{R}}$.

It remains to deal with models of arithmetic:

1. *standard models*
   A *model of arithmetic* is a structure $M = \langle A, R, f_1, f_2, f_3, a\rangle$ such that:

   (a) A is a countable set

   (b) R is a total ordering on A with a first element a and a one-one successor given by f_1, and such that a is not the successor of any element of A, and every element different from a has an immediate predecessor

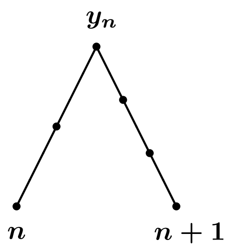
   (c) f_2 and f_3 satisfy the usual recursive definitions of sum and product.

   A *standard model of arithmetic* is a model in which R is a well-ordering.

   To define a standard model, we can use X.6.29 to define an antichain $\{\boldsymbol{c}_n\}_{n\in\omega}$ directly coding the universe A and the ordering R on it, and indirectly coding the graphs of f_1, f_2 and f_3, once they have been represented by appropriate recursive partial orderings. We thus simultaneously build a *single* antichain and a *set* of parameters, defining a set of appropriate relations on elements of the antichain (alternatively, one could code all the needed relations into a single one and build only a single antichain and the parameters definining that relation).

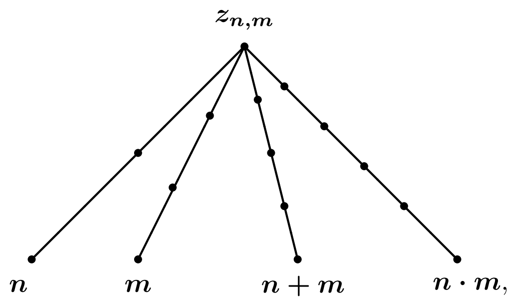
   The universe A will be represented by having, for every n, a degree $\boldsymbol{n}$ of the antichain interpreting the number n.

   The graph of f_1 will be represented by having, for every n, degrees of the antichain coding (through parameters, as in X.6.29) a partial ordering of the following form:



where the number (1 or 2) of elements between $\boldsymbol{y_n}$ and a given $\boldsymbol{i}$ indicates whether the latter corresponds to n or $n+1$.

The graphs of f_2 and f_2 will be similarly represented by having, for every n and m, degrees of the antichain coding a partial ordering of the following form:



where the number (1, 2, 3 or 4) of elements between $\boldsymbol{z_{n,m}}$ and a given $\boldsymbol{i}$ indicates whether the latter corresponds to n, m, $n+m$ or $n \cdot m$.

Given degrees $\vec{\boldsymbol{a}}$ coding a countable antichain of r.e. degrees (intended to be the universe A of a standard model M of arithmetic in $\boldsymbol{\mathcal{R}}$), we can say in a first-order statement of $\boldsymbol{\mathcal{R}}$ that given degrees $\vec{\boldsymbol{c}}$ code a relation and (the graphs of) three functions on the set coded by $\vec{\boldsymbol{a}}$, satisfying the requirements for a model of arithmetic. This is because the definition of a model requires only finitely many first-order properties.

The whole problem is thus to express the fact that the model is standard and this follows the methods of X.5.27. In other words, given any two models of arithmetic M_1 and M_2 coded by parameters $\vec{\boldsymbol{a}}_1$, $\vec{\boldsymbol{c}}_1$ and $\vec{\boldsymbol{a}}_2$, $\vec{\boldsymbol{c}}_2$, we would like to define in a uniform way, with additional parameters $\vec{\boldsymbol{d}}$, a partial map $f : A_1 \to A_2$ that sends the interpretation $\boldsymbol{n}_1$ of n in M_1 to the interpretation $\boldsymbol{n}_2$ of n in M_2.

Then one can say that M_1 coded by $\vec{\boldsymbol{a}}_1$, $\vec{\boldsymbol{c}}_1$ is standard if and only if, for any parameters $\vec{\boldsymbol{a}}_2$, $\vec{\boldsymbol{c}}_2$ coding a model M_2, there are parameters $\vec{\boldsymbol{d}}$ coding a map as above that is total on A_1 and order preserving.

Indeed, if M_1 is standard then, for any M_2, the partial map f defined above is total (because A_1 has only standard elements and f is defined on all of them) and order preserving (because f is then an isomorphism of all of A_1 and the standard part of A_2).

Conversely, if such a map is always total and order preserving, it is so in particular for any standard model M_2. Then M_1 must be standard too, because all the elements of A_1 are mapped in an order preserving way into the elements of A_2, which is standard.

2. *isomorphisms between standard parts of models*
As in X.5.27, to define a map $f : A_1 \to A_2$ providing an isomorphism of the full standard parts of M_1 and M_2, it is enough to uniformly define the finite approximations σ providing isomorphisms of finite initial segments of the standard parts. Then one can define the full (partial) f by saying that $f(\boldsymbol{i}) = \boldsymbol{j}$ if and only if there are degrees $\boldsymbol{x}$ in A_1 and $\boldsymbol{y}$ in A_2, such that $\boldsymbol{i}$ belongs to the initial segment $\mathbf{0}_1, \ldots, \boldsymbol{x}$ of A_1, $\boldsymbol{j}$ belongs to the initial segment $\mathbf{0}_2, \ldots, \boldsymbol{y}$ of A_2, and there is an isomorphism σ of these initial segments such that $\sigma(\boldsymbol{i}) = \boldsymbol{j}$. Then f is the union of a first order definable collection of isomorphisms of initial segments.

To define an isomorphism σ of the initial segments

$$\mathbf{0}_1, \ldots, \boldsymbol{x} \qquad \text{and} \qquad \mathbf{0}_2, \ldots, \boldsymbol{y}$$

of A_1 and A_2, we would like to use X.6.30 to construct an auxiliary standard model M_3 with an initial segment of A_3 whose elements are respectively bounded (in the sense of degrees) by

$$\mathbf{0}_1, \ldots, \boldsymbol{x}, \mathbf{0}_2, \ldots, \boldsymbol{y},$$

with the additional incomparability conditions provided by X.6.30,[6] and such that the remaining elements of A_3 are not bounded by (a given degree bounding all) elements of A_2.

We could then pick the unique element $\boldsymbol{z}$ in A_3 such that

$$\boldsymbol{z} \leq \boldsymbol{x} \ \wedge \ (\boldsymbol{z} +_3 \mathbf{1}_3 +_3 \boldsymbol{z}) \leq \boldsymbol{y},$$

and define σ for $\boldsymbol{i} \leq_1 \boldsymbol{x}$ and $\boldsymbol{j} \leq_2 \boldsymbol{y}$ as

$$\sigma(\boldsymbol{i}) = \boldsymbol{j} \quad \Leftrightarrow \quad (\exists \boldsymbol{s} \in A_3)[\boldsymbol{s} \leq \boldsymbol{i} \ \wedge \ (\boldsymbol{z} +_3 \mathbf{1}_3 +_3 \boldsymbol{s}) \leq \boldsymbol{j}],$$

[6]Notice that here, unlike in the proof of X.5.27, the interpretations of the first integers in M_3 are not *equal* to given degrees of A_1 and A_2, but only *below* them, and thus it does not matter if the two initial segments of A_1 and A_2 have some relations between them.

where $\leq_1$ and $\leq_2$ are the interpretations of the order of the integers in M_1 and M_2 and $+_3$ is the interpretation of the sum of the integers in M_3. This definition of σ uses $\boldsymbol{x}$, $\boldsymbol{y}$ and the parameters needed to define M_3, and we still have to show the existence of an M_3 as above.

But first we check that the definition of σ does produce an isomorphism of the initial segments $\mathbf{0}_1, \ldots, \boldsymbol{x}$ and $\mathbf{0}_2, \ldots, \boldsymbol{y}$, i.e. that

$$\sigma(\boldsymbol{i}) = \boldsymbol{j} \quad \Leftrightarrow \quad (\exists n)[\boldsymbol{i} = \boldsymbol{n}_1 \ \wedge \ \boldsymbol{j} = \boldsymbol{n}_2].$$

In one direction, if $\boldsymbol{i} = \boldsymbol{n}_1$ and $\boldsymbol{j} = \boldsymbol{n}_2$ then $\boldsymbol{s} = \boldsymbol{n}_3$ obviously satisfies the condition of the definition, and so $\sigma(\boldsymbol{i}) = \boldsymbol{j}$.

Conversely, suppose $\sigma(\boldsymbol{i}) = \boldsymbol{j}$ and let $\boldsymbol{s}$ be as in the definition of σ. Since M_3 is a standard model, there is an integer n such that $\boldsymbol{s} = \boldsymbol{n}_3$. We prove that $\boldsymbol{i} = \boldsymbol{n}_1$ and $\boldsymbol{j} = \boldsymbol{n}_2$.

- By construction of M_3, there is exactly one element below $\boldsymbol{i}$ in the initial segment of A_3 bounded by

  $$\mathbf{0}_1, \ldots, \boldsymbol{x}.$$

  Since $\boldsymbol{s} = \boldsymbol{n}_3$, then $\boldsymbol{i} = \boldsymbol{n}_1$.
- By construction of M_3, any element of A_3 (such as $\boldsymbol{z} +_3 \mathbf{1}_3 +_3 \boldsymbol{s}$) bounded by an element of A_2 (such as $\boldsymbol{j}$), must be in the initial segment bounded by

  $$\mathbf{0}_1, \ldots, \boldsymbol{x}, \mathbf{0}_2, \ldots, \boldsymbol{y}.$$

  Moreover, since $\boldsymbol{z} +_3 \mathbf{1}_3 +_3 \boldsymbol{s}$ follows $\boldsymbol{z}$ in A_3, by the first part of the proof it must actually be in the segment of A_3 bounded by

  $$\mathbf{0}_2, \ldots, \boldsymbol{y}.$$

  But, by construction of M_3, there is exactly one element below $\boldsymbol{j}$ in such segment. Since $\boldsymbol{s} = \boldsymbol{n}_3$, then $\boldsymbol{j} = \boldsymbol{n}_2$.

3. *good parameters*
The strategy sketched above would work if we knew how to construct auxiliary models in general. This can easily be done by using X.6.30, but only if the r.e. degrees that we want the interpretations of the integers to avoid are low. This produces an additional complication, since then the reasoning used above to define standard models requires a quantification over *low* parameters coding given models, and the notion of lowness is not definable in $\boldsymbol{\mathcal{R}}$ (see p. 581).

However, it is not really lowness that we need, but only the appropriate consequences we can derive from it. Namely, to be able to define embeddings between models. And this can be defined directly, as follows.

A degree $\boldsymbol{g}$ is called *good* for a model M_2 coded by parameters $\vec{\boldsymbol{a}}_2$, $\vec{\boldsymbol{c}}_2$ if:

- the elements of M_2 are below $\boldsymbol{g}$
- for any model M_1 coded by parameters $\vec{\boldsymbol{a}}_1$, $\vec{\boldsymbol{c}}_1$ and whose elements are below $\boldsymbol{g}$, there are parameters coding a map $f : A_1 \to A_2$ as above that is total and order preserving.

The construction of X.6.30 ensures that there exist standard models coded by parameters below any given low r.e. degree, thus showing in particular that any such degree is good. The class of standard models coded by parameters below some good degree is thus not empty, and the same reasoning as above then shows that any model coded by parameters below a good degree is standard.

Since the notion of goodness (relative to a given model) is obviously definable, this solves the problem. $\square$

Corollary X.6.32 (Harrington and Shelah [1982]) *The first-order theory of $\mathcal{R}$ is undecidable and not axiomatizable.*

Harrington and Shelah [1982] originally obtained the corollary by a $\mathbf{0}'''$-priority argument, showing that all Δ^0_2 partial orderings are uniformly definable in the r.e. degrees from parameters, via a coding on degrees satisfying a maximality property of the following kind:

$$\boldsymbol{x} \text{ is a maximal degree below } \boldsymbol{c} \text{ not joining } \boldsymbol{a} \text{ above } \boldsymbol{b}.$$

Slaman and Woodin simplified the proof of the corollary, bringing it down to a still complicated type of $\mathbf{0}''$-priority argument, via a coding on degrees satisfying a dual minimality property of the following kind:

$$\boldsymbol{x} \text{ is a minimal degree below } \boldsymbol{c} \text{ joining } \boldsymbol{a} \text{ above } \boldsymbol{b},$$

Ambos-Spies and Shore [1993] produced a yet simpler proof, by using the priority method needed to build branching and nonbranching r.e. degrees (see X.6.8), to show that all finite partial orderings are uniformly definable with parameters in the r.e. degrees, via a coding satisfying a maximality property of the following kind:

$$\boldsymbol{x} \text{ is a maximal degree above } \boldsymbol{c} \text{ meeting some } \boldsymbol{y} \text{ at } \boldsymbol{c}.$$

Nies, Shore and Slaman [1998] finally gave the proof above, which uses only the priority method needed to build incomparable r.e. degrees and minimal pairs, via the Slaman and Woodin coding.

As for the full result X.6.31, the original proof of Harrington and Slaman via the Harrington and Shelah coding, as well as the tamer one of Slaman and Woodin via their own coding, not only were much more complicated than the corresponding proofs of undecidability alone, but also used an indirect approach that interpreted the theory of First-Order Arithmetic without directly defining a standard model of arithmetic in $\mathcal{R}$, unlike that of Nies, Shore and Slaman [1998] given above.

Nies [1998] has adapted the latter proof to show that *the first-order theory of $\mathcal{R}_{wtt}$ has the same degree (and actually the same isomorphism type) as the theory of First-Order Arithmetic*, and in particular it is undecidable and not axiomatizable (Ambos-Spies, Nies and Shore [1992]).

Turning to subclasses of sentences, X.1.9 shows that the one-quantifier theory of $\mathcal{R}$ is decidable, with the same decision procedure as for $\mathcal{D}$. Slaman and Soare [1995] have solved the extension of embeddings problem (see p. I.490), thus showing that *a fragment of the two-quantifier theory of $\mathcal{R}$ is decidable*, but it is not known whether the full two-quantifier theory is as well. On the other hand, Lempp, Nies and Slaman [1998] have proved that *the three-quantifier theory of $\mathcal{R}$ is undecidable*.

For the r.e. *wtt*-degrees, one has that *the two-quantifier theory of $\mathcal{R}_{wtt}$ is decidable* (Fejer and Shore [1985], Ambos-Spies, Fejer, Lempp and Lerman [1996]), and *the four-quantifier theory of $\mathcal{R}_{wtt}$ is undecidable* (Lempp and Nies [1995]). It is not known whether the three-quantifier theory of $\mathcal{R}_{wtt}$ is undecidable (the methods used to prove the undecidability of the three-quantifier theory of $\mathcal{R}$ are highly non distributive, and thus do not apply to $\mathcal{R}_{wtt}$ because of X.6.27.c).

Absolute definability $\star$

First we note that, since $\mathcal{R}$ is itself definable in $\mathcal{D}$ (Cooper [1990]), everything definable in $\mathcal{R}$ is also definable in $\mathcal{D}$. The converse does not hold. For example, **I** is definable in $\mathcal{D}$ (since so is any relation definable in Second-Order Arithmetic and invariant under double jump, by the results quoted after XI.5.17), but not in $\mathcal{R}$ (see below).

Looking at individual r.e. degrees, $\mathbf{0}$ and $\mathbf{0}'$ are obviously definable in $\mathcal{R}$ as, respectively, the least and the greatest elements. No other r.e. degree is known to be definable, but certainly not everyone is (see X.6).

Looking at classes of r.e. degrees, the following general definability result (a local version of V.7.12) has been obtained by Nies, Shore and Slaman [1998], by a combination of the coding methods used to prove X.6.31 and XI.5.4: *a* *relation on r.e. degrees that is invariant under double jump is definable in $\mathcal{R}$ if and only if it is definable in First-Order Arithmetic.* From this one obtains, with an additional argument for $\mathbf{H}_1$, that *all jump classes* $\mathbf{L}_n$ *and* $\mathbf{H}_n$ *are definable in* $\mathcal{R}$*, with the only exception of* $\mathbf{L}_1$. Since $\mathbf{L}_1$ *is not definable* (see below), it is not possible to improve the general definability result to relations invariant under jump. On the other hand *the jump class* $\mathbf{I}$ *is not definable in* $\mathcal{R}$ because, by X.9.18.c, it is not even definable in arithmetic.

No nontrivial definability result is known for $\mathcal{R}_{wtt}$.

Homogeneity $\star$

We can look at $\mathcal{R}$ both as the principal ideal $\mathcal{R}(\geq \mathbf{0}')$ and as the principal filter $\mathcal{R}(\leq \mathbf{0}')$, i.e. as the upper cone above $\mathbf{0}$ and the lower cone below $\mathbf{0}'$.

Proposition X.6.33 Failure of homogeneity for $\mathcal{R}$. *Homogeneity fails for $\mathcal{R}$, both on upper cones (even with low bases) and on lower cones (even with high bases).*

Proof. Minimal pairs exist in the r.e. degrees (X.6.5), but not in the r.e. degrees above a nonbranching degree. Failure of homogeneity on upper cones (with low bases) thus follows from the existence of (low) nonbranching degrees (X.6.8.d).

The Diamond Theorem fails in the r.e. degrees (X.6.23), but it holds in the r.e. degrees below the join of a minimal pair. Failure of homogeneity on lower cones (with high bases) thus follows from the existence of (high) minimal pairs (X.6.7.b). $\square$

Homogeneity fails for $\mathcal{R}_{wtt}$, too. The proof above works for the second part but not for the first, since every r.e. *wtt*-degree is branching (X.6.8.a). In this case one can however use the existence of r.e. *wtt*-degrees having g.l.b. and joining to $\mathbf{0}'$ (Downey [1989a]), so that the Diamond Theorem holds in the r.e. *wtt*-degrees above the g.l.b.

We do not know whether there are elementarily equivalent intervals of $\mathcal{R}$, but we can easily prove the following.

Proposition X.6.34 (Shore [1993]) *There are isomorphic intervals of $\mathcal{R}_{wtt}$.*

Proof. We show that, for any r.e. *wtt*-degrees $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ such that $\boldsymbol{a} \cap \boldsymbol{b} = \boldsymbol{c}$, the map

$$f(\boldsymbol{x}) = \boldsymbol{x} \cup \boldsymbol{b}$$

provides an isomorphism of the intervals $[\boldsymbol{c}, \boldsymbol{a}]$ and $[\boldsymbol{b}, \boldsymbol{a} \cup \boldsymbol{b}]$.

- *f is a homomorphism*

$$f(\boldsymbol{x}) \cup f(\boldsymbol{y}) = (\boldsymbol{x} \cup \boldsymbol{b}) \cup (\boldsymbol{y} \cup \boldsymbol{b}) = (\boldsymbol{x} \cup \boldsymbol{y}) \cup \boldsymbol{b} = f(\boldsymbol{x} \cup \boldsymbol{y}).$$

- *f is one-one*

  Given $\boldsymbol{x}, \boldsymbol{y} \in [\boldsymbol{c}, \boldsymbol{a}]$, suppose $f(\boldsymbol{x}) = f(\boldsymbol{y})$, i.e. $\boldsymbol{x} \cup \boldsymbol{b} = \boldsymbol{y} \cup \boldsymbol{b}$. Since $\boldsymbol{x} \leq \boldsymbol{y} \cup \boldsymbol{b}$, by distributivity there are $\boldsymbol{y_0} \leq \boldsymbol{y}$ and $\boldsymbol{b_0} \leq \boldsymbol{b}$ such that $\boldsymbol{x} = \boldsymbol{y_0} \cup \boldsymbol{b_0}$. It follows from $\boldsymbol{x} \leq \boldsymbol{a}$ that $\boldsymbol{b_0} \leq \boldsymbol{a}$, and from $\boldsymbol{a} \cap \boldsymbol{b} = \boldsymbol{c}$ that $\boldsymbol{b_0} \leq \boldsymbol{c}$. Then

$$\boldsymbol{x} = \boldsymbol{y_0} \cup \boldsymbol{b_0} \leq \boldsymbol{y_0} \cup \boldsymbol{c},$$

  and from $\boldsymbol{c} \leq \boldsymbol{y}$ we have $\boldsymbol{x} \leq \boldsymbol{y}$. Symmetrically $\boldsymbol{y} \leq \boldsymbol{x}$. So $\boldsymbol{x} = \boldsymbol{y}$.

- *f is onto*

  Given $\boldsymbol{z} \in [\boldsymbol{b}, \boldsymbol{a} \cup \boldsymbol{b}]$, by distributivity there are $\boldsymbol{a_0} \leq \boldsymbol{a}$ and $\boldsymbol{b_0} \leq \boldsymbol{b}$ such that $\boldsymbol{z} = \boldsymbol{a_0} \cup \boldsymbol{b_0}$ and hence $\boldsymbol{y} = \boldsymbol{a_0} \cup \boldsymbol{b}$. Then

$$f(\boldsymbol{a_0} \cup \boldsymbol{c}) = (\boldsymbol{a_0} \cup \boldsymbol{c}) \cup \boldsymbol{b} = \boldsymbol{a_0} \cup \boldsymbol{b} = \boldsymbol{z},$$

  and $(\boldsymbol{a_0} \cup \boldsymbol{c}) \in [\boldsymbol{c}, \boldsymbol{a}]$. □

One can also look at 'external' homogeneity, by considering the structure $\boldsymbol{\mathcal{R}}^{\boldsymbol{a}}$ of the degrees r.e. in and above a given degree $\boldsymbol{a}$. By building on relativized versions of the definability results quoted above Nies, Shore and Slaman [1998], improving on Shore [1982b], have obtained the following local versions of V.7.13 and V.7.16:

- *if $\boldsymbol{\mathcal{R}}^{\boldsymbol{a}}$ is elementarily equivalent to $\boldsymbol{\mathcal{R}}$, then $\boldsymbol{a}$ is low$_2$*
- *if $\boldsymbol{\mathcal{R}}^{\boldsymbol{a}}$ is isomorphic to $\boldsymbol{\mathcal{R}}^{\boldsymbol{b}}$, then $\boldsymbol{a}$ and $\boldsymbol{b}$ have the same double-jump.*

Automorphisms ⋆

Before we discuss the existence of automorphisms of $\boldsymbol{\mathcal{R}}$, we consider automorphism bases.

Proposition X.6.35 (Sacks [1963], Lerman) *For any $n \geq 1$, both $\mathbf{L}_n$ and $\mathbf{H}_n$ are automorphism bases for $\mathcal{R}$.*

Proof. It is enough to prove the result for $n = 1$.

To prove that $\mathbf{L}_1$ is an automorphism base, it is enough to show that it generates the r.e. degrees. This follows from X.6.15.a, which says that every nonzero r.e. degree is the l.u.b. of two low r.e. degrees.

To prove that $\mathbf{H}_1$ is an automorphism base, we show that if f is an automorphism that moves an r.e. degree $\boldsymbol{a}$, i.e. $f(\boldsymbol{a}) \neq \boldsymbol{a}$, then f also moves a high r.e. degree $\boldsymbol{b}$. By possibly considering f^{-1}, we may suppose that $f(\boldsymbol{a}) \not\leq \boldsymbol{a}$. As in XI.1.13, there is an incomplete high r.e. degree $\boldsymbol{b}$ such that $\boldsymbol{a} \leq \boldsymbol{b}$ and $f(\boldsymbol{a}) \not\leq \boldsymbol{b}$. Then $f(\boldsymbol{a}) \leq f(\boldsymbol{b})$ because f is an automorphism, and so $f(\boldsymbol{b}) \not\leq \boldsymbol{b}$, and hence $f(\boldsymbol{b}) \neq \boldsymbol{b}$. $\square$

Ambos-Spies [199?] has proved that *every nontrivial lower cone of r.e. degrees is an automorphism base for* $\boldsymbol{\mathcal{R}}$.

Exercises X.6.36 Generators of $\boldsymbol{\mathcal{R}}$.

a) $\mathbf{H_1}$ *does not generate* $\boldsymbol{\mathcal{R}}$. (Ambos-Spies [1985b]) (Hint: given any nonzero r.e. degree $\boldsymbol{a}$, there is an r.e. degree $\boldsymbol{b} \leq \boldsymbol{a}$ that cannot be obtained as a finite meet of r.e. degrees. Then $\boldsymbol{b}$ can only be generated as a join of elements below it, and hence below $\boldsymbol{a}$. Then the degrees not below $\boldsymbol{a}$ cannot generate the r.e. degrees. This slightly generalizes X.6.8.d, which built an r.e. degree below $\boldsymbol{a}$ that is not the meet of two r.e. degrees.)

b) *A jump class generates* $\boldsymbol{\mathcal{R}}$ *if and only if it contains* $\mathbf{L_1}$. (Ambos-Spies [1985b]) (Hint: see part a).)

c) $\boldsymbol{\mathcal{R}}$ *is generated by two proper principal ideals.* (Welch) (Hint: by X.6.15.a, consider a low Sacks splitting A and B of $\mathcal{K}_0 = \bigoplus_{e\in\omega} \mathcal{W}_e$. Since $\mathcal{W}_e$ is the e-th column of $\mathcal{K}_0$, $A^{[e]} \cup B^{[e]} = \mathcal{W}_e$, $A^{[e]} \leq_T A$ and $B^{[e]} \leq_T B$.)

d) *No finite number of proper principal filters generates* $\boldsymbol{\mathcal{R}}$. (Hint: see the proof of part a).)

We now turn to automorphisms. Cooper [1997] has proved that *there are at least countably many automorphisms of* $\boldsymbol{\mathcal{R}}$, but their exact number is not known. Although Slaman and Woodin [199?] have proved that $\boldsymbol{\mathcal{D}}$ can only have countably many automorphisms, the result does not immediately translate to $\boldsymbol{\mathcal{R}}$ because not every automorphism of $\boldsymbol{\mathcal{R}}$ can be extended to one of $\boldsymbol{\mathcal{D}}$ (see below).

Since $\mathbf{L}_1$ is an automorphism base by X.6.35, any automorphism of $\boldsymbol{\mathcal{R}}$ moves a low r.e. degree and thus *not every low r.e. degree is definable in* $\boldsymbol{\mathcal{R}}$. By building not only an automorphism of $\boldsymbol{\mathcal{R}}$, but one that sends a low r.e. degree into a nonlow one, Cooper [1997] has proved that $\mathbf{L}_1$ *is not definable in* $\boldsymbol{\mathcal{R}}$.

$\mathbf{L}_1$ is definable in $\boldsymbol{\mathcal{D}}$ by the definability of $\boldsymbol{\mathcal{R}}$ (Cooper [1990]) and of the jump operator (XI.5.17), and hence it must be fixed under every automorphism of $\boldsymbol{\mathcal{D}}$. Thus, any automorphism of $\boldsymbol{\mathcal{R}}$ that does not fix $\mathbf{L}_1$ cannot be the restriction to $\boldsymbol{\mathcal{R}}$ of an automorphism of $\boldsymbol{\mathcal{D}}$. This proves that *not every automorphism of* $\boldsymbol{\mathcal{R}}$ *can be extended to one of* $\boldsymbol{\mathcal{D}}$.

In the opposite direction, it is known that $\boldsymbol{\mathcal{R}}$ *is an automorphism base for* $\boldsymbol{\mathcal{D}}$ (Slaman and Woodin [199?]). Together with the definability of $\boldsymbol{\mathcal{R}}$ in $\boldsymbol{\mathcal{D}}$

(Cooper [1990]), this shows that *every nontrivial automorphism of $\mathcal{D}$ induces a nontrivial one of $\mathcal{R}$.*

Both the existence of at most countably many automorphisms of $\mathcal{R}$, and the characterization of the relations of r.e. degrees definable in $\mathcal{R}$ as those that are both invariant under automorphisms of $\mathcal{R}$ and definable in First-Order Arithmetic, would follow from a proof of the so-called *biinterpretability with parameters of $\mathcal{R}$ and the standard model of arithmetic*, i.e. the possibility of defining in $\mathcal{R}$ parameters not only a standard model of arithmetic (as in X.6.31), but also the map taking an r.e. degree $\boldsymbol{x}$ into a set (of natural numbers in the standard model) of degree $\boldsymbol{x}$ (a strong local version of V.7.10). Then each automorphism would be determined by the image of the parameters, and any definition in arithmetic of a relation on degrees could be translated into a definition in the degrees through the definable map (as in V.7.11).

X.7 Comparison of Degree Theories $\star$

In this section we consider other notions of degree introduced in Chapter III, namely **r.e. 1-degrees**, ***btt*-degrees**, and ***tt*-degrees**. As in Section VI.5, we will mostly quote results and content ourselves to develop the various theories only to the point needed to show that they are not elementarily equivalent among themselves, nor to the r.e. Turing and m-degrees. This will be done mostly by looking at minimal degrees.

For strong reducibilities, the analogues of the results from X.5.1 to X.5.7 either hold with the same proof (for reducibilities between m and tt) or follow from the results for m-degrees (for 1-degrees). Some of these results were originally proved separately, by Kallibekov [1973], [1973a], Degtev [1973], and Ladner and Sasso [1975]. Degtev's papers [1979], [1979a], [1981], [1982], [1983], [1983a] and [1985] give additional information on strong reducibilities.

One-one degrees

We know from Section V.5 that the (r.e.) 1-degrees have features that distinguish them from all other degrees. We were able in III.7.3 to define a degree $\mathbf{0}_1$, but this had some peculiar properties. First, below it there are infinitely many 1-degrees (those containing finite or cofinite sets). Second, there are also infinitely many r.e. 1-degrees incomparable with it (by III.7.4, the 1-degrees of simple sets).

We could certainly restrict our attention to the r.e. 1-degrees between $\mathbf{0}_1$ and $\mathbf{0}'_1$, but this seems of limited interest (although it does give, as quoted below, the undecidability of the theory). Some results (e.g. X.5.1) hold by the same proof as for r.e. m-degrees, since $\mathbf{0}'_1 = \mathbf{0}'_m$ by III.7.5. We add here a new result.

Definition X.7.1 *A 1-degree* $\boldsymbol{a}$ *is* **minimal** *if it is greater than* $\mathbf{0}_1$*, and there is no 1-degree strictly between* $\boldsymbol{a}$ *and* $\mathbf{0}_1$*.*

Proposition X.7.2 (Lachlan [1969]) *There exists a minimal r.e. 1-degree.*

Proof. Let A be a maximal set and let B and C be two recursively inseparable r.e. sets such that $A = B \cup C$ (IX.2.5). We show that the 1-degree of B is minimal.

The 1-degree of B is above $\mathbf{0}_1$. Indeed, given an infinite and coinfinite recursive set R and two recursive one-one enumerations f of B and g of C, then $R \leq_1 B$ via the function

$$h(x) = \begin{cases} f(n) & \text{if } x \text{ is the } n\text{-th element of } R \\ g(n) & \text{if } x \text{ is the } n\text{-th element of } \overline{R}. \end{cases}$$

By the proof of X.5.13.a, the m-degree of B is minimal and it does not contain simple sets. Let D be an r.e. nonrecursive set such that $D \leq_1 B$. Then $D \leq_m B$ and, by minimality of the m-degree of B, $B \leq_m D$. We want to show that $B \leq_1 D$.

Since D is not simple (being in the m-degree of B), and is infinite and coinfinite (being nonrecursive), there are two recursive infinite sets $D_1 \subseteq D$ and $D_2 \subseteq \overline{D}$. If

$$x \in D \Leftrightarrow f(x) \in B$$

and S is the range of f, let g be the recursive function so defined:

- if x shows up first in B, let $g(x)$ be the least element of D_1 which is not in $\{g(0), \ldots, g(x-1)\}$
- if x shows up first in C or $x \in \overline{S} \cap \overline{A}$, let $g(x)$ be the least element of D_2 which is not in $\{g(0), \ldots, g(x-1)\}$
- if x shows up first in S, we would like to let $g(x)$ be the least z such that $f(z) = x$. We can do this if $z \notin \{g(0), \ldots, g(x-1)\}$, but in the opposite case g would not be one-one. But since $f(z)$ is unique, it is enough to note that if $z \in \{g(0), \ldots, g(x-1)\}$, then it must be so because of one of the two clauses above, hence either $z \in D_1 \subseteq D$ or $z \in D_2 \subseteq \overline{D}$, and it is then enough to take as $g(x)$ an element of D_1 or D_2, respectively, which is not in $\{g(0), \ldots, g(x-1)\}$.

Notice that $\overline{S} \cap \overline{A}$ is finite. Indeed, $f(\overline{D})$ cannot be finite, otherwise D would be recursive (since $x \in \overline{D}$ if and only if $f(x) \in f(\overline{D})$). Then $f(\overline{D}) = S \cap \overline{A}$ is infinite, and by maximality $\overline{S} \cap \overline{A}$ is finite.

It follows that g is recursive and $B \leq_1 D$ via g by construction. □

Exercises X.7.3 a) *Every nonzero r.e. T-degree contains a minimal r.e. 1-degree.* (Degtev [1976]) (Hint: modify the construction of X.5.20.a as follows. Let each block contribute one box whenever $\mathcal{W}_e$ is intersected, for every e. By considering $\mathcal{W}_e = \omega$ we see that almost every equivalence class of $\overline{B}$ is infinite. By considering $\mathcal{W}_{g(x)} = \{y : x\eta y\}$ and applying VI.6.3 we see that B is a cylinder. Now let C be r.e. nonrecursive, and $C \leq_m B$. We show below that C is also a cylinder. Since B has minimal m-degree, $C \equiv_m B$. Since every set in the m-degree of B is a cylinder, the m-degree consists of just one 1-degree, and then $B \leq_1 C$ and B has minimal 1-degree.

To show that C is a cylinder, let f reduce C to B, and let $\mathcal{W}_e$ be its range. Define

$$x\eta^* y \;\Leftrightarrow\; f(x)\eta f(y).$$

Since C is η^*-closed by definition of f, it is enough to show that almost every equivalence class of $\overline{C}$ is infinite. Since $\overline{C} = f^{-1}(\mathcal{W}_e \cap \overline{B})$ and C is nonrecursive, the η-closure of $\mathcal{W}_e \cap \overline{B}$ is not r.e. But then it must be η-infinite, and by construction it must be constant on $\overline{B}$ from a certain point on. This means that, by the variation of the construction proposed at the beginning, $\mathcal{W}_e$ intersects almost every equivalence class of B infinitely often. By taking counterimages via f, almost every equivalence class of $\overline{C}$ is infinite.)

b) *In the 1-degrees between $\mathbf{0}_1$ and $\mathbf{0}_1'$, for any pair $\boldsymbol{b}$ and $\boldsymbol{c}$ of r.e. 1-degrees such that $\mathbf{c} < \mathbf{0}_m'$ and $\mathbf{c} \not\leq \boldsymbol{b}$, there is a strong minimal cover of $\boldsymbol{b}$ in the r.e. 1-degrees incomparable with $\mathbf{c}$.* (Hint: in the proof of X.5.24 the coding is already one-one. Since every set whose 1-degrees is above $\mathbf{0}_1'$ is immune and coimmune, the m-reductions can be made one-one by the technique used in X.7.2. Finally, combining strategies produces only finitely many mistakes, and the 1-degrees above $\mathbf{0}_1'$ are closed under finite modifications.)

Lachlan [1969] has proved that every finite initial segment of the r.e. 1-degrees (above $\mathbf{0}_1$) is a distributive lattice. It follows from part b) of the exercises, as in X.5.25, that *the topped finite initial segments of $\boldsymbol{\mathcal{R}_1}$ are exactly the distributive lattices*. In particular, being $\mathbf{0}_1$ definable in $\boldsymbol{\mathcal{R}_1}$ (see p. I.584), *the first-order theory of $\boldsymbol{\mathcal{R}_1}$ is undecidable and not axiomatizable*.

The initial segments of the r.e. 1-degrees have not yet been classified. We also do not know whether the r.e. 1-degrees above $\mathbf{0}_1$ are distributive (the non-r.e. 1-degrees are not, by Lachlan [1969]).

Turning to r.e. 1-degrees in general, we have the following result.

Proposition X.7.4 *$\boldsymbol{\mathcal{R}_1}$ is neither an upper nor a lower semilattice, i.e. l.u.b. and g.l.b. do not always exist.*

Proof. The proof of VI.5.1 actually produced r.e. sets without l.u.b. and g.l.b. in the r.e. 1-degrees. □

Notice that the r.e. sets without l.u.b. and g.l.b. in the 1-degrees produced by the proof of VI.5.1 are actually simple, in particular they are incomparable with $\mathbf{0}_1$. We do not know whether the r.e. 1-degrees above $\mathbf{0}_1$ are an uppersemilattice.

Bounded truth-table degrees

It will be proved in XI.5.9.a that there are two r.e. sets A and B such that the Turing degree of $A - B$ does not contain any r.e. set. In particular, since $A - B \leq_{btt} \mathcal{K}$, *$\boldsymbol{\mathcal{R}_{btt}}$ is not an ideal of $\boldsymbol{\mathcal{D}_{btt}}$*, and the same holds for any reducibility between $\leq_{btt}$ and $\leq_T$ (in contrast to the r.e. m-degrees). Thus, the next result is nontrivial.

Proposition X.7.5 (Kobzev [1973]) *If an r.e. btt-degree is minimal in $\boldsymbol{\mathcal{R}_{btt}}$, it is also minimal in $\boldsymbol{\mathcal{D}_{btt}}$.*

Proof. It is enough to show that if A is r.e. and $B \leq_{btt} A$ is nonrecursive, there is a nonrecursive r.e. set C such that either $C \leq_m B$ or $C \leq_m \overline{B}$, so that $C \leq_{btt} B$.

Let

$$x \in B \ \Leftrightarrow\ A \models \sigma_{f(x)}.$$

We showed on p. I.332 that there are elements $\{b_1^x, \dots, b_n^x\}$ such that the answer to the question 'is x in B?' depends solely on the membership of the b_i^x's in A. Let

$$x \in P \ \Leftrightarrow\ \{b_1^x, \dots, b_n^x\} \subseteq A.$$

Then $P \subseteq B$ or $P \subseteq \overline{B}$. Let

$$x \in P_i \ \Leftrightarrow\ \{b_1^x, \dots, \hat{b_i^x}, \dots, b_n^x\},$$

where $\hat{b_i^x}$ means that we drop the element b_i^x. Note that one of $B \cap P_i$ and $\overline{B} \cap P_i$ is always r.e., since membership in one of them is either independent of b_i^x, or decided only by the fact that $b_i^x \in A$.

- If, for some i, one of $B \cap P_i$ and $\overline{B} \cap P_i$ is not r.e., consider it. For example, let $B \cap P_i$ be not r.e. Then $\overline{B} \cap P_i$ is r.e. If P_i is the range of the recursive function g, let

  $$x \in C \ \Leftrightarrow\ g(x) \in \overline{B} \cap P_i.$$

  Then C is r.e. by definition, and it is not recursive because $B \cap P_i = g(\overline{C})$ is not r.e. Finally, $C \leq_m \overline{B}$ because $g(x)$ is always in P_i, hence

  $$x \in C \ \Leftrightarrow\ g(x) \in \overline{B}.$$

  The other case is treated similarly, and it produces a set C such that $C \leq_m B$.

- If both $B \cap P_i$ and $\overline{B} \cap P_i$ are r.e. for every i, consider

$$x \in P_{i,j} \;\Leftrightarrow\; \{b_1^x, \ldots, \hat{b_i^x}, \ldots, \hat{b_j^x}, \ldots, b_n^x\} \subseteq A.$$

  Again, one of $B \cap P_{i,j}$ and $\overline{B} \cap P_{i,j}$ is r.e. We can then proceed as above, and so on.

- If in the end all the $\hat{P}_i$ defined as

$$x \in \hat{P}_i \;\Leftrightarrow\; b_i^x \in A$$

  are such that $B \cap \hat{P}_i$ and $\overline{B} \cap \hat{P}_i$ are r.e., then one of B and $\overline{B}$ is also r.e. (and we can set C equal to it). For example, suppose

$$\{b_1^x, \ldots, b_n^x\} \subseteq \overline{A} \;\Rightarrow\; x \in B.$$

  Then

$$x \in \overline{B} \;\Rightarrow\; \{b_1^x, \ldots, b_n^x\} \cap A \neq \emptyset.$$

  To determine whether x is in B wait until, for some i, $b_i^x \in A$, so that $x \in \hat{P}_i$. Then generate $B \cap \hat{P}_i$ and $\overline{B} \cap \hat{P}_i$, and see whether $x \in B$ or $x \in \overline{B}$. Thus $\overline{B}$ is r.e. $\Box$

We turn now to existence results.

Proposition X.7.6 (Kobzev [1973]) *A simple set of minimal bd-degree has minimal btt-degree.*

Proof. We prove that if A is simple and B is a nonrecursive r.e. set such that $B \leq_{btt} A$, then there is a nonrecursive r.e. set C such that $C \leq_{bd} A$ and $C \leq_{btt} B$. If A has minimal *bd*-degree, then $A \leq_{bd} C$ and so $A \leq_{btt} B$. Let

$$x \in B \;\Leftrightarrow\; A \models \sigma_{f(x)},$$

with $\{b_1^x, \ldots, b_n^x\}$ the elements used in $\sigma_{f(x)}$. Let

$$x \in P \;\Leftrightarrow\; \{b_1^x, \ldots, b_n^x\} \subseteq A.$$

Then $P \subseteq B$ or $P \subseteq \overline{B}$. We proceed by induction on n.

1. $P \subseteq \overline{B}$
   Then

$$\{b_1^x, \ldots, b_n^x\} \subseteq A \;\Rightarrow\; x \in \overline{B},$$

   and so

$$x \in B \;\Rightarrow\; \{b_1^x, \ldots, b_n^x\} \cap \overline{A} \neq \emptyset.$$

There is a finite set $F \subseteq \overline{A}$ such that

$$x \in B \ \Rightarrow \ \{b_1^x, \dots, b_n^x\} \cap F \neq \emptyset,$$

otherwise we can obtain an infinite r.e. sequence of n-tuples intersecting $\overline{A}$, contradicting simplicity (see the proof of III.8.8). Consider the recursive set

$$x \in R \ \Leftrightarrow \ \{b_1^x, \dots, b_n^x\} \cap F \neq \emptyset.$$

Then $B \subseteq R$, $P \subseteq \overline{R}$ and $\overline{B} \cap R$ is not r.e. (otherwise B is recursive as follows: to see if $x \in B$, first see if $x \in R$, and if so generate B and $\overline{B} \cap R$).

Consider the recursive sets

$$x \in R_i \ \Leftrightarrow \ b_i^x \in F.$$

Then $R = \bigcup_{1 \leq i \leq n} R_i$. Since $\overline{B} \cap R$ is not r.e., one $\overline{B} \cap R_i$ is not r.e. The set $B_1 = B \cup \overline{R}_i$ is then nonrecursive and r.e. Moreover:

- $B_1 \leq_m B$ via
$$h(x) = \begin{cases} b & \text{if } x \in \overline{R}_i \\ x & \text{otherwise,} \end{cases}$$
where $b \in B$.
- $B_1 \leq_{btt} A$ with a reduction of norm $n - 1$. Indeed, we can dispense with b_i^x in $\sigma_{f(x)}$ as follows. If $b_i^x \notin F$, then $x \in B_1$ (as $\overline{R}_i \subseteq B_1$). If $b_i^x \in F$, then $b_i^x \in \overline{A}$ (as $F \subseteq \overline{A}$).

We can thus apply the induction hypothesis to B_1.

2. $P \subseteq B$
Then
$$\{b_1^x, \dots, b_n^x\} \subseteq A \ \Rightarrow \ x \in B,$$
and so
$$x \in \overline{B} \ \Rightarrow \ \{b_1^x, \dots, b_n^x\} \cap \overline{A} \neq \emptyset.$$
Consider
$$x \in P_i \ \Leftrightarrow \ \{b_1^x, \dots, \hat{b_i^x}, \dots, b_n^x\} \subseteq A.$$
We have the following cases:

(a) *for some i, $\overline{B} \cap P_i$ is not r.e.*
Let g be a recursive function with range P_i, and
$$x \in C \ \Leftrightarrow \ g(x) \in B \cap P_i.$$

Then C is an r.e. nonrecursive set. Also, $C \leq_m B$ because

$$x \in C \Leftrightarrow g(x) \in B,$$

since $g(x) \in P_i$. Moreover, $C \leq_m A$ because

$$x \in C \Leftrightarrow b_i^{g(x)} \in A.$$

Indeed, $g(x) \in P_i$, and so membership of $g(x)$ in B is determined only by $b_i^{g(x)}$. Now, by case hypothesis ($P \subseteq B$),

$$b_i^{g(x)} \in A \Rightarrow g(x) \in P \Rightarrow g(x) \in B \Rightarrow x \in C,$$

and we may suppose

$$b_i^{g(x)} \notin A \Rightarrow g(x) \notin B$$

(otherwise $x \in C$, independently of $b_i^{g(x)}$, and we can dispense with it in the reduction).

(b) *for every i, $\overline{B} \cap P_i$ is r.e.*
Consider the sets $F_i = \{b_i^x : x \in \overline{B} \cap P_i\}$. They are all r.e. and contained in $\overline{A}$, since

$$x \in \overline{B} \Rightarrow \{b_1^x, \ldots, b_n^x\} \cap \overline{A} \neq \emptyset$$

and

$$x \in P_i \Rightarrow \{b_1^x, \ldots, \hat{b_i^x}, \ldots, b_n^x\} \subseteq A,$$

so

$$x \in \overline{B} \cap P_i \Rightarrow b_i^x \in \overline{A}.$$

Since A is simple, each F_i is finite. We can thus consider the recursive sets

$$x \in R_i \Leftrightarrow b_i^x \in F_i$$

(note that $\overline{B} \cap P_i \subseteq R_i$ by definition), and again there are two cases.

i. *for some i, $\overline{B} \cap R_i$ is not r.e.*
Let h be a recursive function with range R_i, and

$$x \in B_1 \Leftrightarrow h(x) \in B \cap R_i.$$

Then B_1 is a nonrecursive r.e. set, $B_1 \leq_m B$ via h, and $B_1 \leq_{btt} A$ with norm $n-1$ (since

$$x \in B_1 \Leftrightarrow h(x) \in B,$$

but $h(x) \in R_i$ and so $b_i^{h(x)} \in F_i \subseteq \overline{A}$). We can thus apply the induction hypothesis again.

ii. *for every i, $\overline{B} \cap R_i$ is r.e.*
Now consider $B_1 = B \cup (\bigcup_{1 \le i \le n} R_i)$. Then B_1 is a nonrecursive r.e. set, otherwise B would be recursive as follows: to see if $x \in B$, first see if $x \in B_1$, and if so generate B and the $\overline{B} \cap R_i$. Also, $B_1 \le_m B$. And

$$x \in \overline{B}_1 \;\Rightarrow\; |\{b_1^x, \ldots, b_n^x\} \cap \overline{A}| \ge 2.$$

Indeed, if $x \in \overline{B}_1$, then $x \in \overline{B}$ and $\{b_1^x, \ldots, b_n^x\} \cap \overline{A} \ne \emptyset$. Given any $b_i^x \in \overline{A}$, the set $\{b_1^x, \ldots, \hat{b_i^x}, \ldots, b_n^x\}$ cannot be contained in A, otherwise $x \in P_i$, so $x \in \overline{B} \cap P_i$ and $x \in R_i$, hence $x \in B_1$, which contradicts the hypothesis.
But then we can start case 2 again, with B_1 in place of B and $P_{i,j}$ in place of P_i. For example, in case 2.a we have $\overline{B}_1 \cap P_{i,j}$ not r.e., and if C is defined as there, then $C \le_d A$ since

$$x \in C \;\Leftrightarrow\; \{b_i^{g(x)}, b_j^{g(x)}\} \cap A \ne \emptyset,$$

and so on.

Since the only case that does reduce to the induction hypothesis is 2.a, at the end we obtain $C \le_{bd} A$ with norm $\le n$. □

Corollary X.7.7 *A semirecursive r.e. set of minimal m-degree has minimal btt-degree.*

Proof. Let A be an r.e. semirecursive set of minimal m-degree. We may suppose that A is simple. Otherwise, let $D \le_d A$ be the simple (deficiency) set obtained as in II.6.16. By semirecursiveness of A, $D \le_m A$. By minimality, $D \equiv_m A$. By III.5.6, D is semirecursive.

Now let B be an r.e. nonrecursive set such that $B \le_{btt} A$. Since A is simple, the proof of X.7.6 produces an r.e. set C such that $C \le_{bd} A$ and $C \le_{btt} B$. By semirecursiveness of A, $C \le_m A$; by minimality, $A \equiv_m C$. Thus $A \equiv_m C \le_{btt} B$, i.e. $A \le_{btt} B$, and A has minimal *btt*-degree. □

Kobzev [1973] has shown that *maximal sets do not have minimal btt-degree.* Since they do have minimal m-degree, one cannot improve the results above by having 'minimal m-degree' in place of 'minimal *bd*-degree' in the proposition, or by dropping 'semirecursive' in the corollary.

Corollary X.7.8 (Degtev [1973]) *There exists an r.e. set of minimal (r.e.) btt-degree. Actually, a nonrecursive η-maximal semirecursive set is such.*

Proof. By III.5.19, nonrecursive η-maximal semirecursive sets exist. By X.5.12, they have minimal m-degree. By X.7.7, they have minimal r.e. *btt*-degree, and by X.7.5 also minimal *btt*-degree. $\Box$

Exercises X.7.9 a) *Every nonzero r.e. T-degree contains an r.e. set of minimal btt-degree.* (Kobzev [1973a]) (Hint: given a nonrecursive r.e. set B, we build a simple set A of minimal *bd*-degree and such that $B \equiv_T A$. The proof is an extension of X.5.20.a, to which we refer. In particular, $B \equiv_T A$ is ensured as there.

To obtain A of minimal *bd*-degree we act as we would to obtain a minimal m-degree, but now considering strong arrays in place of recursive functions. When

$$z \in C \;\Leftrightarrow\; D_{g(z)} \cap A \neq \emptyset$$

and C is nonrecursive, then we want all boxes A_n^s from a certain point on to contain some $D_{g(x)}$. Thus, when at stage $s+1$ we see a box A_x^s (in a given block) which does not contain any $D_{\varphi_{e,s}(z)}$ for some $e \leq x$, but such that there is some $D_{\varphi_{e,s}(y)} \subseteq \overline{A}_s$ which only intersects boxes in following blocks, then we collect these boxes and send them all into A_x, which is then going to contain $D_{\varphi_e(y)}$ after this stage.

A is automatically simple. Otherwise, let C and D be two infinite recursive subsets of $\overline{A}$, and

$$D_{g(z)} = \begin{cases} a & \text{if } z \in C \\ z & \text{if } z \in D \\ b & \text{otherwise,} \end{cases}$$

with $a \in A$ and $b \in \overline{A}$. Then $C \leq_{bd} A$ via g, and infinitely many $D_{g(z)}$'s are contained in $\overline{A}$. The construction thus gives $C \equiv_{bd} A \equiv_T B$, but B is not recursive while C is.

We still have to choose f in such a way that in the n-th block there is always a nonempty box. Suppose we already have $f(n)$. For each $x < f(n)$ we might want to put into A_x the elements of $D_{\varphi_e(y)}$, for some y and $e \leq x$. We may suppose that $D_{\varphi_e(y)}$ has at most e elements, otherwise we modify the array. Since we are only interested in bounded reductions, and every recursive function has infinitely many indices, this causes no problem. Thus for each $e \leq x$ we need at most e elements, and for each x we need $\sum_{e \leq x} e = \frac{x(x+1)}{2}$ elements. Moreover, we need one more block for the coding, and another one to ensure that $\overline{A}$ is η-infinite. We may thus define $f(0) = 0$ and $f(n+1) = f(n) + \sum_{x<f(n)} \frac{x(x+1)}{2} + 2$.)

b) *Every nonzero r.e. T-degree contains infinitely many minimal r.e. btt-degrees.* (Kobzev [1973a]) (Hint: modify part a) by using priority and permitting.)

While the use of simple sets of minimal *bd*-degree for the construction of minimal r.e. *btt*-degrees is elegant, one sometimes needs more flexibility. In the next result we give *a direct construction of a minimal r.e. btt-degree* and use it to prove that such degrees are unbounded.

Theorem X.7.10 (Marchenkov [1977]) *For every r.e. btt-degree $\boldsymbol{c}$ such that $\mathbf{0}_{btt} < \boldsymbol{c} < \mathbf{0}'_{btt}$ there is a minimal r.e. btt-degree $\boldsymbol{a}$ incomparable with it.*

Proof. Given an r.e. set C nonrecursive and such that $\mathcal{K} \not\leq_{btt} C$, we want to build an r.e. set A of minimal *btt*-degree such that $A \not\leq_{btt} C$. The proof is a modification of that for X.5.21. We will have to use boxes instead of single elements, so we use recursive boxes of boxes

$$S_\sigma = \{X_0^\sigma < X_1^\sigma < \cdots\},$$

where $X_n^\sigma < X_{n+1}^\sigma$ means that $(\max\ X_n^\sigma) < (\min\ X_{n+1}^\sigma)$. Moreover, for each σ, the cardinality of X_n^σ is bounded.

The requirement

$$P_e\ :\ A \not\leq_{btt} C \text{ via } \varphi_e$$

is treated in the usual way, namely by coding $\mathcal{K}$ into

$$Q_\sigma = \{X_0^\sigma < X_2^\sigma < \cdots\},$$

for $e = |\sigma|$. Here we use

$$l_p(e,s) = \max\,\{z : (\forall y < z)[y \in A_s \ \Leftrightarrow\ C_s \models \sigma_{\varphi_{e,s}(y)}]\},$$

all computations converging. If $A \leq_{btt} C$ via φ_e, then $\lim_{s\to\infty} l_p(e,s) = \infty$ and we will have

$$n \in \mathcal{K} \ \Leftrightarrow\ X_{2n}^\sigma \subseteq A,$$

so that $\mathcal{K} \leq_{btt} A \leq_{btt} C$, which is a contradiction.

We now show how to treat the 0-th minimality requirement. Suppose that

$$x \in B \ \Leftrightarrow\ A \models \sigma_{\varphi_e(x)}$$

with bounded tables. We have to take action on

$$R_\sigma = \{X_1^\sigma < X_3^\sigma < \cdots\}.$$

As in the case of m-degrees, we try to build S_0 in such a way that if it is infinite, then we have $A \leq_{btt} B$. In X.5.21 we simply had to ensure that every element of S_0 is in $\mathcal{W}_0$, since then we could use the m-reduction of B to A. We now have to ensure that if

$$S_0 = \{X_0^0 < X_1^0 < \cdots\},$$

then

$$X_n^0 \subseteq A \ \Leftrightarrow\ B \text{ satisfies some } btt\text{-condition.}$$

Suppose at stage $s+1$ we have that, for some x:

1. $\varphi_{0,s}(x)$ converges
2. the set $\{b_1^x, \ldots, b_p^x\}$ of elements of $\sigma_{\varphi_0(x)}$ is contained in $\overline{A}_s \cap S_1$

3. the Boolean function obtained by identifying in $\sigma_{\varphi_0(x)}$ elements which are in the same box of S_1 is not constant, i.e. there is a 0-splitting on x

4. the smallest element in the boxes $X^1_{n_0}, \ldots, X^1_{n_i}$ containing some b^x_j is greater than max $S_{0,s}$.

Since the Boolean function in part 3 is not constant, there is an effective way of sending some of these boxes $X^1_{n_0}, \ldots, X^1_{n_i}$ into A, and collecting the others into a single box X^* in such a way that

$$X^* \subseteq A \quad \Leftrightarrow \quad x \in B. \tag{X.2}$$

We send this box X^* into S_0, and the boxes of S_1 with index $\leq n_j$ for some j into A.

To fulfill the condition that S_0 consists of boxes of bounded cardinality, it is enough to use the convention that the e-th reduction uses at most $e+1$ elements (since we are only interested in bounded reductions, and every recursive function has infinitely many indices, this causes no problem).

If S_0 is infinite, we obtain $A \leq_{btt} B$ because $Q_\emptyset$ only contributes finitely to A, $S_1 \subseteq A$, and for each box of S_0 we can find the corresponding x such that X.2 holds.

If S_0 is finite, then we can argue that the truth-table reduction of B to A is not bounded. Indeed, only for finitely many x are the conditions above satisfied. If φ_0 is total, then for almost every x:

- either the set $\{b^x_1, \ldots, b^x_p\}$ intersects A or S_0 or $Q_\emptyset$, so we can find one element in it (note that S_0 is finite in the present case, and so are both $S_0 \cap A$ and $Q_\emptyset \cap A$)

- or the Boolean function considered in 3 above is constant (and then actually no element is needed).

We leave to the reader the task of spelling out the global construction, in which strategies are, as usual, played one inside the other. □

Corollary X.7.11 Capping Theorem for $\mathcal{R}_{btt}$. *Every nontrivial r.e. btt-degree is part of a minimal pair.*

Fenner and Schaefer [1999] and Downey [199?] have proved the **Non-Cupping Theorem for $\mathcal{R}_{btt}$**, in the following strong form: *the btt-degree of a simple set is not part of a pair joining to $\mathbf{0}'_{btt}$.*

Exercises X.7.12 a) **Diamond Theorem for $\mathcal{R}_{btt}$** (Jockusch and Mohrherr [1985]) *There is a minimal pair of r.e. btt-degrees joining to $\mathbf{0}'_{btt}$.* (Hint: see X.7.20.a.)

b) *$\boldsymbol{\mathcal{R}_{btt}}$ is not distributive.* (Degtev [1979]) (Hint: see X.7.20.b.)

Nies [1992] (see also Nies [1997]) has proved that *the first-order theory of $\boldsymbol{\mathcal{R}_{btt}}$ is undecidable and not axiomatizable*, but it is not known whether it has the same degree as the theory of First-Order Arithmetic.

Turning to subclasses of sentences, Nies [1996] has proved that *the four-quantifier theory of $\boldsymbol{\mathcal{R}_{btt}}$ is undecidable.*

Truth-table degrees

The following is a simple lemma that has interesting applications.

Proposition X.7.13 (Degtev [1972], [1978b]) *If A is a simple semirecursive set and B is a nonrecursive (not necessarily r.e.) set such that $B \leq_{tt} A$, then $B \equiv_T A$.*

Proof. By III.5.4, A is the lower cut of a recursive linear ordering $\sqsubseteq$ of ω. If $x \in \overline{A}$, then $\{y : x \sqsubseteq y\}$ is a recursive subset of $\overline{A}$, and by simplicity it is finite. To enumerate $\overline{A}$ it is thus enough to enumerate (the upper closure of) an infinite subset of it. If we obtain such an enumeration recursively in B, then $\overline{A}$ is r.e. in B and hence (being A r.e.) $A \leq_T B$.

If $x \in \overline{A}$ is given, we show how to obtain a new element of $\overline{A}$. Let

$$z \in B \Leftrightarrow A \models \sigma_{f(z)}.$$

Compute recursively the function $h(z)$ that gives the Boolean value of $\sigma_{f(z)}$ under the assignment that associates 0 with the elements y such that $x \sqsubseteq y$ (since such elements are in $\overline{A}$) and 1 to the others. If it were

$$z \in B \Leftrightarrow h(z) = 1$$

for every z, B would be recursive. Thus there is a z for which this fails, and it can be found recursively in B. Since

$$z \in B \Leftrightarrow A \models \sigma_{f(z)} \qquad \text{but} \qquad z \in B \Leftrightarrow h(z) \neq 1,$$

the assignment above must be wrong. Thus it gives the wrong value to some $y \sqsubset x$ used in $\sigma_{f(z)}$. In other words, some such $y \sqsubset x$ is in $\overline{A}$. Since $\overline{A}$ is closed upwards w.r.t. $\sqsubseteq$, the greatest $y \sqsubset x$ (w.r.t. $\sqsubseteq$) used in $\sigma_{f(z)}$ is in $\overline{A}$. □

One can now obtain *an alternative construction of a minimal pair of r.e. tt-degrees* as follows. Let A_0 and A_1 be simple semirecursive sets (e.g. nondeficiency sets, see II.6.16). If $B \leq_{tt} A_0, A_1$ and B is nonrecursive, then $B \equiv_T A_0$ and $B \equiv_T A_1$. Thus, if A_0 and A_1 have different T-degrees,

$$B \leq_{tt} A_0, A_1 \Rightarrow B \text{ recursive},$$

and they form a minimal pair for *tt*-degrees.

Exercises X.7.14 (Nies) a) *Every T-incomplete r.e. tt-degree is part of a minimal pair*. (Hint: let X be any T-incomplete r.e. set, and A be a T-complete, simple, and semirecursive set, which exists by II.6.16. Then A and X form a minimal pair in the *tt*-degrees, because if $B \leq_{tt} A, X$ were not recursive, then $B \equiv_T A$ by X.7.13, hence $A \leq_T B \leq_{tt} X$, and X would be T-complete.)

b) *Some T-complete r.e. tt-degree is part of a minimal pair*. (Hint: by part a).)

c) *Every T-incomplete r.e. tt-degree is branching*. (Hint: let X and Y be any T-incomplete r.e. sets such that $X <_{tt} Y$, and A be a simple semirecursive set such that $A \not\leq_T Y$. To show that $A \oplus X$ and Y have X as g.l.b., suppose $B \leq_{tt} A \oplus X, Y$. By a relativization of X.7.13, if it were $B \not\leq_{tt} X$ then $B \oplus X \equiv_T A$, hence

$$A \leq_T B \oplus X \leq_{tt} Y \oplus X \equiv_{tt} Y,$$

and we would have $A \leq_T Y$.)

d) *Every nonrecursive r.e. tt-degree that is least among the r.e. tt-degrees in its own T-degree is a minimal tt-degree*. (Hint: let C be a nonrecursive r.e. set having least r.e. *tt*-degree in its own T-degree, and A be a simple semirecursive set such that $A \equiv_T C$. Then $C \leq_{tt} A$ by the hypothesis on C. Suppose B is a nonrecursive set such that $B \leq_{tt} C$. Then $B \leq_{tt} A$, hence $B \equiv_T A$ by X.7.13, and $C \leq_{tt} B$ again by the hypothesis on C. Thus C has minimal *tt*-degree.)

Parts a) and b) are optimal, since by X.7.19 not every *tt*-incomplete r.e. *tt*-degree is part of a minimal pair. Part c) can instead be improved to the **Branching Theorem for $\boldsymbol{\mathcal{R}_{tt}}$**, i.e. *every tt-incomplete r.e. tt-degree is branching* (Fejer and Shore [199?]).

Proposition X.7.15 (Kobzev [1978]) *If A is a nonrecursive, semirecursive η-maximal set, and B is a (not necessarily r.e.) set such that $B \leq_T A$, then either B is recursive or $A \leq_{tt} B$.*

Proof. We prove the following: *if A is a nonrecursive, semirecursive and η-hyperhypersimple set and $D \equiv_T A$, then there is a nonrecursive r.e. set C such that $C \leq_m A$ and $C \leq_{tt} D$.* From this we obtain the stated result as follows. First remember that if A is also η-maximal, then it has minimal m-degree (X.5.12). Thus $D \equiv_T A$ implies $A \leq_{tt} D$ (since $A \equiv_m C \leq_{tt} D$). By II.6.16 there is a simple semirecursive set A_1 such that $A_1 \equiv_T A$. By letting $D = A_1$, $A \leq_{tt} A_1$. So if $B \leq_{tt} A$, then $B \leq_{tt} A_1$. By X.7.13, if B is nonrecursive we obtain $B \equiv_T A_1$ and hence $B \equiv_T A$. By letting $D = B$, $A \leq_{tt} B$.

We now prove the claim. First note that we may suppose:

- *A simple*
  Otherwise there is $A_1 \equiv_T A$ simple semirecursive such that $A_1 \leq_d A$ (II.6.16). Since A is semirecursive, $A_1 \leq_m A$. If

$$x \in A_1 \;\Leftrightarrow\; g(x) \in A$$

and A is η-hyperhypersimple, then A_1 is η_1-hyperhypersimple, where

$$x\eta_1 y \Leftrightarrow g(x)\eta g(y).$$

Thus if $C \leq_m A_1$, then $C \leq_m A$.

- *D semirecursive*
  Otherwise let $D_1 \equiv_{tt} D$ be semirecursive (III.5.5.a). Thus if $C \leq_{tt} D_1$, then $C \leq_{tt} D$.

Since $A \equiv_T D$, A is r.e. and both A and D are semirecursive, we have (III.1.4 and III.5.2) R and Q r.e. such that

$$\begin{array}{lcl} x \in \overline{A} & \Leftrightarrow & (\exists u, v)[R(x,u,v) \ \wedge \ u \in D \ \wedge \ v \in \overline{D}] \\ c_D(x) = y & \Leftrightarrow & (\exists a)[Q(x,y,a) \ \wedge \ a \in \overline{A}], \end{array}$$

so that

$$x \in \overline{A} \Leftrightarrow (\exists u, v, a_0, a_1)[R(x,u,v) \wedge a_0, a_1 \in \overline{A} \wedge Q(u,1,a_1) \wedge Q(v,0,a_0)].$$

Since A is semirecursive, it is the lower cut of a recursive linear ordering $\sqsubseteq$. We may suppose that

$$x\eta z \wedge x \sqsubseteq y \sqsubseteq z \Rightarrow x\eta y,$$

otherwise just consider the transitive and symmetric closure of

$$x\eta^* y \Leftrightarrow x\eta y \vee (\exists z)(x \sqsubseteq y \sqsubseteq z \wedge x\eta z).$$

If A is η-hyperhypersimple, then it also η^*-hyperhypersimple. Indeed, if A is simple and co-η-infinite, then each η-class of $\overline{A}$ is finite, and hence so is each η^*-class. Thus an array for η^* would produce an array for η.

Define the following r.e. approximation to $\overline{A}$:

$$\begin{array}{lcl} x \in P & \Leftrightarrow & (\exists u, v, a_0, a_1, b)[R(x,u,v) \wedge Q(u,1,a_1) \wedge Q(v,0,a_0) \wedge \\ & & x\eta b \wedge b \sqsubseteq a_0, a_1]. \end{array}$$

There are two cases:

1. *$\overline{A} \cap P$ infinite*
   From this we obtain C as required, as follows. Let h be a recursive function with range P, and define

   $$x \in C \Leftrightarrow h(x) \in A.$$

   Note that $h(\overline{C}) = \overline{A} \cap P$.

C must be nonrecursive, otherwise $\overline{C}$ is r.e. and so is $\overline{A} \cap P$. However, by case hypothesis this is an infinite r.e. subset of $\overline{A}$, contradicting simplicity.

$C \leq_m A$ by definition, and $C \leq_{tt} D$ as follows. Note that

$$x \in \overline{C} \Leftrightarrow h(x) \in \overline{A}.$$

Since $h(x)$ is always a member of P, let $u(x)$, $v(x)$, $a_0(x)$, $a_1(x)$ and $b(x)$ be recursive functions which associate with $h(x)$ elements u, v, a_0, a_1 and b as in the definition of P. We obtain $C \leq_{tt} D$ from

$$x \in \overline{C} \Leftrightarrow u(x) \in D \wedge v(x) \in \overline{D}.$$

- If $u(x) \in D$ and $v(x) \in \overline{D}$, then by definition $h(x) \in \overline{A}$ and so $x \in \overline{C}$.
- If $x \in \overline{C}$, then $h(x) \in \overline{A}$. By definition of P, $h(x)\eta b(x)$ and so $b(x) \in \overline{A}$. From $b(x) \sqsubseteq a_0(x), a_1(x)$ we have $a_0(x), a_1(x) \in \overline{A}$. So $u(x) \in D$ and $v(x) \in \overline{D}$.

2. $\overline{A} \cap P$ *finite*

We show that A is not η-hyperhypersimple, and hence this case cannot occur. Define $\mathcal{W}_{f(x)}$ as follows. For every a_0, a_1 such that

$$(\exists u, v)[R(x,u,v) \wedge Q(u,1,a_1) \wedge Q(v,0,a_0)],$$

we choose the element between a_0 and a_1 which is smallest w.r.t. $\sqsubseteq$ (call it a) and put it into $\mathcal{W}_{f(x)}$. There are two cases:

- $x \in A$
  a_0 and a_1 cannot both be in $\overline{A}$ (otherwise $x \in \overline{A}$), so $a \in A$ (because a is the smallest between a_0 and a_1 w.r.t. $\sqsubseteq$). Hence $\mathcal{W}_{f(x)} \subseteq A$.
- $x \in \overline{A}$
  Since $\overline{A} \cap P$ is finite, we may suppose $x \in \overline{P}$. If $a \in \mathcal{W}_{f(x)}$, from $x\eta x$ it follows that $a \sqsubset x$ (otherwise $x = b$ gives $x \in P$), and from $a \sqsubseteq a_1, a_2$ it follows that $\neg(x\eta a)$ (otherwise $a = b$ gives $x \in P$).

By the usual technique (see III.4.6) we may suppose that $|\mathcal{W}_{f(x)} \cap \overline{A}| \leq 1$. Then, for some $x_0 \in \overline{A}$, let

$$\begin{aligned} \mathcal{W}_{g(0)} &= \{x_0\} \\ \mathcal{W}_{g(n+1)} &= \bigcup_{z \in \mathcal{W}_{g(n)}} \mathcal{W}_{f(z)}. \end{aligned}$$

By induction it is easy to see that $|\mathcal{W}_{g(n)} \cap \overline{A}| = 1$. Also, if x_n is the unique element of $\mathcal{W}_{g(n)} \cap \overline{A}$,

$$x_n \sqsubset x_{n+1} \qquad \text{and} \qquad \neg(x_n \eta x_{n+1}).$$

By what is proved above, the x_n are all in $\overline{A}$ (because x_0 is) and in different classes of equivalence w.r.t. η. Thus the η-closures of the $\mathcal{W}_{g(n)}$ witness that A is not η-hyperhypersimple. $\square$

Corollary X.7.16 (Degtev [1973], Marchenkov [1975], Kobzev [1978]) *There exists an r.e. set of minimal (r.e.) tt-degree. Actually, a nonrecursive η-maximal semirecursive set is such.*

The corollary shows that an r.e. tt-degree minimal in $\boldsymbol{\mathcal{R}_{tt}}$ can be minimal also in $\boldsymbol{\mathcal{D}_{tt}}$. It is not known whether this is always the case, as it is for the btt-degrees (see X.7.5).

Exercises X.7.17 a) *Every r.e. set of minimal r.e. tt-degree is low$_2$.* In particular, *maximal sets do not have minimal r.e. tt-degree.* (Degtev [1978b], Downey and Shore [1995]) (Hint: given C r.e. and not low$_2$, build an r.e. set A such that $C \leq_T A$ (in particular, A is not recursive), $A \leq_{tt} C$, and $C \not\leq_{tt} A$, so that C does not have minimal r.e. tt-degree.

Since C is not recursive, it is not tt-reducible to any cofinite set. Thus, for any e and x, if $\overline{X}$ has at most x elements then there is an n such that the e-th tt-reduction

$$z \in C \;\Leftrightarrow\; X \models \sigma_{\varphi_e(z)}$$

fails on an initial segment σ of X of length n, and hence fails as well for any set extending σ, i.e. having at most x elements up to n. As in the proof of IV.2.6.b there is a fixed n such that, if X has at most x elements up to n, then C is not tt-reducible to X via φ_e; moreover, the least such n can be found recursively in $\mathcal{K}$. If we then define $h(x)$ as the maximum of such n's, for $e \leq x$, we have $h \leq_T \mathcal{K}$.

Since C is not low$_2$, by XI.1.10 there is a function $f \leq_T C$ not dominated by h, say $f \simeq \{e\}^C$. Since C is r.e., f can be recursively approximated, for example by $f_1(x,s) = \{e\}_s^{C_s}(x)$.

The construction of A is as follows. At stage $s+1$, if $\overline{A}_s = \{a_0^s < a_1^s < \cdots\}$, and if either $f_1(x,s)$ has changed or C has changed below $f_1(x,s)$, put in A all elements between a_x^s and $\max\{f_1(x,s), s\}$. If $\overline{A} = \{a_0 < a_1 < \cdots\}$, then $a_x \geq a_x^{s+1} \geq f_1(x,s) \geq f(x)$. Hence, as a function of x, a_x is not dominated by h, because such is f.

$C \leq_T A$ because, to obtain longer and longer initial segments of C, it is enough to run the construction of A until a_x has reached its final value. Then $f_1(x,s)$ no longer changes and C no longer changes below $f_1(x,s)$.

To see that $A \leq_{tt} C$, given y wait for a stage s such that $y \leq s$. If $y \in A_s$, there is nothing to do. Otherwise, y will go into A if and only if some $a_x^s \leq y$ will too, and this will happen only if $f_1(x,s)$ will change, or C will change below $f_1(x,s)$. The various possibilities can be described by a truth-table.

To see that $C \not\leq_{tt} A$, notice that there are infinitely many x such that $a_x > h(x)$, because a_x is not dominated by h. But below any such a_x there are at most x elements of A, by definition of a_x; then C cannot be tt-reducible to A via φ_e for any $e \leq x$, by definition of h.)

b) *Every η-maximal semirecursive set is low$_2$.* (Miller [1981] (Hint: by X.7.16 and part a). See IX.2.31 for a different proof.)

c) *There exists an r.e. nonzero T-degree that contains exactly one minimal r.e. tt-degree.* (Kobzev [1978]) (Hint: if A is a nonrecursive η-maximal semirecursive set and $A \equiv_T B$, then $A \leq_{tt} B$ by X.7.15.)

The results above show that the minimal r.e. *tt*-degrees are not as abundant as the minimal r.e. *m*-degrees or *btt*-degrees (see X.5.20 and X.7.9). A complete characterization of the r.e. T-degrees containing minimal r.e. *tt*-degrees is not known.

Parts a) and b) cannot be improved, since Miller [1981] has shown that there are low$_2$ but not low$_1$ η-maximal semirecursive sets, and hence low$_2$ but not low$_1$ minimal r.e. *tt*-degrees.

The next results contrast with X.5.21 and X.7.10.

Proposition X.7.18 (Marchenkov [1976a]) *The minimal r.e. tt-degrees are bounded below $\mathbf{0}'_{tt}$.*

Proof. By the proof of II.6.16, every minimal r.e. *tt*-degree contains an r.e. coretraceable set. By II.6.19, there is a recursive enumeration $\{A_e\}_{e\in\omega}$ of the class of such sets. Let

$$B_e = A_0 \oplus \cdots \oplus A_e.$$

Then $\mathcal{K} \not\leq_{tt} B_e$ because each such set is the join of recursive or hypersimple sets. By the Thickness Lemma X.3.7 there is an r.e. set C such that $B_e \leq_{tt} C$ and $\mathcal{K} \not\leq_{tt} C$. C is then an incomplete r.e. set bounding a representative of each minimal r.e. *tt*-degree. $\square$

Corollary X.7.19 Non-Capping Theorem for $\boldsymbol{\mathcal{R}_{tt}}$ (Degtev) *There are nontrivial r.e. tt-degrees that are not part of a minimal pair.*

Proof. By upward density (see X.5.2), it follows from the proof of the result above that there is an r.e. *tt*-degree $\boldsymbol{a} < \mathbf{0}'_{tt}$ which is strictly above all minimal r.e. *tt*-degrees and all the r.e. *tt*-degrees containing nondeficiency sets. Let $\boldsymbol{b}$ be any r.e. *tt*-degree incomparable with it. Since $\boldsymbol{b} \not\leq \boldsymbol{a}$, $\boldsymbol{b}$ is not minimal. Thus there is an r.e. *tt*-degree $\boldsymbol{c}$ such that $\mathbf{0}_{tt} < \boldsymbol{c} < \boldsymbol{b}$. By II.6.16, $\boldsymbol{c}$ bounds a nonzero r.e. *tt*-degree $\boldsymbol{d}$ containing a nondeficiency set. Then $\boldsymbol{d} \leq \boldsymbol{c} < \boldsymbol{b}$, and $\boldsymbol{d} < \boldsymbol{a}$ by the choice of $\boldsymbol{a}$. Thus $\boldsymbol{a}$ and $\boldsymbol{b}$ do not form a minimal pair, and $\boldsymbol{a}$ is not part of a minimal pair. $\square$

The proof above produces a T-complete r.e. *tt*-degree that is not part of a minimal pair, because there are T-complete nondeficiency sets. This is necessary, since by X.7.14.a every T-incomplete r.e. *tt*-degree is part of a minimal pair.

Downey [1988a] has proved the **Non-Cupping Theorem for $\boldsymbol{\mathcal{R}_{tt}}$**, i.e. *there are nontrivial r.e. tt-degrees that are not part of a pair joining to $\mathbf{0}'_{tt}$.*

Exercises X.7.20 Lattice embeddings.

a) **Diamond Theorem for $\boldsymbol{\mathcal{R}_{tt}}$** (Jockusch and Mohrherr [1985]) *There is a minimal pair of r.e. tt-degrees joining to* $\mathbf{0}'_{tt}$. (Hint: we want low r.e. sets A and B such that $A \cup B = \mathcal{K}$. Then $A \oplus B \equiv_{tt} \mathcal{K}$ (in one direction, $A \oplus B \leq_m \mathcal{K}$ follows from the fact that A and B are r.e.). Also, A and B are nonrecursive, being low and joining to $\mathcal{K}$. The requirements are (see X.3.3)

$$\begin{array}{lcl} P & : & x \in \mathcal{K} \Rightarrow x \in A \vee x \in B \\ N_e^A & : & (\exists_\infty s)(\{e\}_s^{A_s}(e)\downarrow) \Rightarrow \{e\}^A(e)\downarrow \\ N_e^B & : & (\exists_\infty s)(\{e\}_s^{B_s}(e)\downarrow) \Rightarrow \{e\}^B(e)\downarrow . \end{array}$$

The negative requirements are satisfied by the usual finite restraint. Moreover, A and B have to be a minimal pair, and so we have also the requirements

$$R_{i,j} \ : \ (\forall x)[x \in C \Leftrightarrow A \models \sigma_{\varphi_i(x)} \Leftrightarrow B \models \sigma_{\varphi_j(x)}] \Rightarrow C \text{ recursive.}$$

The usual strategy for minimal pairs (of preserving one side of the agreement) does not work here, since it produces restraints on both sides A and B. But when an element is generated in $\mathcal{K}$, we have to put it into one of A and B, to satisfy P.

The strategy for $R_{i,j}$ is the following. If x is generated in $\mathcal{K}$ at stage s, see if introducing it into A or B modifies some computation, i.e. for some $z \leq s$

$$(\forall y < z)[A_s \models \sigma_{\varphi_i(y)} \Leftrightarrow B_s \models \sigma_{\varphi_j(y)}]$$

but either

$$A_s \models \sigma_{\varphi_i(z)} \quad \Leftrightarrow \quad A_s \cup \{x\} \not\models \sigma_{\varphi_i(z)} \tag{X.3}$$

or

$$B_s \models \sigma_{\varphi_j(z)} \quad \Leftrightarrow \quad B_s \cup \{x\} \not\models \sigma_{\varphi_j(z)}. \tag{X.4}$$

If not, we can put x into A or B, without problems for $R_{i,j}$. Otherwise, suppose e.g. that X.3 holds. Then we can satisfy $R_{i,j}$ with finite restraint, by forcing a disagreement as follows: put x into B (to satisfy P), and then choose whether to put x into A or not, in such a way as to ensure that

$$A_{s+1} \models \sigma_{\varphi_i(z)} \Leftrightarrow B_{s+1} \not\models \sigma_{\varphi_j(z)}.$$

This can be done since, by X.3, putting x into A changes the truth value of the left-hand-side. Similarly if X.4 holds.)

b) $\boldsymbol{\mathcal{R}_{tt}}$ *is not distributive.* (Degtev [1979]) (Hint: it is enough to embed the pentagon. We want low r.e. sets A, B and C such that $C <_{tt} A$, $B \oplus C \equiv_{tt} \mathcal{K}$, and A and B for a minimal pair in the *tt*-degrees. Then B and C are automatically nonrecursive. To have $C \leq_{tt} A$, put $2x$ into A whenever x goes into C. $A \not\leq_{tt} C$ is ensured as usual.)

Jockusch and Mohrherr [1985], Fejer and Shore [1985] and Haught [1987] have proved that *every finite (actually, every recursively presented) lattice is embeddable* *in $\mathcal{R}_{tt}$, preserving least and greatest elements.*

While the use of η-maximal semirecursive sets for the construction of minimal r.e. *tt*-degrees (X.7.16) is elegant, one sometimes needs more flexibility. To give *a direct construction of a minimal r.e. tt-degree* one can adapt either the method of boxes used in X.7.10 for minimal r.e. *btt*-degrees (see X.7.21.a), or the method of trees used in V.5.11 for minimal T-degrees (see Fejer and Shore [1990]). Building on this, Haught and Shore [1990] have embedded many lattices as initial segments of $\boldsymbol{\mathcal{R}_{tt}}$. Harrington and Haught [199?] have proved that *not every finite distributive lattice is embeddable as an initial segment of* $\boldsymbol{\mathcal{R}_{tt}}$, e.g. the diamond is not (more generally, every finite initial segment of $\boldsymbol{\mathcal{R}_{tt}}$ is a lattice above a minimal degree). A complete characterization of the initial segment of $\boldsymbol{\mathcal{R}_{tt}}$ is not known.

Exercises X.7.21 a) *Give a direct construction of a minimal r.e. tt-degree.* (Hint: as in III.5.19, by letting the e-states monitor the i-splittings of the first element of each box w.r.t. *tt*-reductions, for each $i \leq e$. The construction amalgamates boxes whose first element is not e-splitting to boxes whose first element is, so that an e-splitting box never follows a box that is not e-splitting. If $B \leq_{tt} A$ via the e-th reduction, then either almost all boxes are not e-splitting and B is recursive, or all boxes are e-splitting and one can inductively define a *tt*-reduction of A to B, using semirecursiveness.)

b) *Every non-low$_2$ r.e. T-degree contains an r.e. tt-degree without minimal predecessors.* (Degtev [1973]) (Hint: by II.6.16, any r.e. T-degree contains a simple semirecursive set A. By X.7.13, if $B \leq_{tt} A$, then either B is recursive or $B \equiv_T A$, i.e. every nonrecursive predecessor of A has the same T-degree as A. But if A is nonlow$_2$ then, by X.7.17.a, its T-degree does not contain minimal r.e. *tt*-degrees.)

c) *Every initial segment of r.e. tt-degrees above a minimal r.e. tt-degree consists only of sets belonging to a single low$_2$ T-degree.* (Nies and Shore [1995]) (Hint: let B be a set in the minimal r.e. *tt*-degree of the initial segment, and X be a set in any nonrecursive r.e. *tt*-degree of the initial segment. By the proof of II.6.16, there is a simple semirecursive set such that $A \equiv_T X$ and $A \leq_{tt} X$. Since X belongs to the initial segment, so does A. Then $B \leq_{tt} A$ by the minimality of B, and $B \equiv_T A$ by X.7.13, so that $B \equiv_T X$. Thus all *tt*-degrees in the initial segment belong to the T-degree of B, which is low$_2$ by X.7.17.a.)

Part c) holds in particular for every *finite* initial segment of r.e. *tt*-degrees, since Harrington and Haught [199?] have proved that every such initial segment is above a minimal degree.

Nies and Shore [1995] have proved that *an ideal of $\boldsymbol{\mathcal{R}_{tt}}$ consisting only of T-incomplete tt-degrees has an exact pair if and only if it is bounded below $\mathbf{0}'_{tt}$ and Σ^0_3.* It is not known whether the result holds in general, without the T-incompleteness condition.

Nies and Shore [1995] have proved that *the first-order theory of* $\boldsymbol{\mathcal{R}_{tt}}$ *has the same degree (and, actually, the same isomorphism type) as the theory of First-Order Arithmetic*, and in particular it is undecidable and not axiomatizable (Haught and Shore [1990]).

Turning to subclasses of sentences, Fejer and Shore [1985] have solved the extension of embeddings problem (see p. I.490), thus showing that the one-quantifier theory, as well as *a fragment of the two-quantifier theory of* $\boldsymbol{\mathcal{R}_{tt}}$ *is decidable*, but it is not known whether the full two-quantifier theory is as well. On the other hand, Nies [1996] has proved that *the three-quantifier theory of* $\boldsymbol{\mathcal{R}_{tt}}$ *is undecidable.*

Elementary inequivalences

What we have proved until now allows us to state the main result of this section, on the comparison of r.e. degrees theories.

Theorem X.7.22 (Young [1964], Lachlan [1966], Degtev [1973], Ladner and Sasso [1975], Marchenkov [1976a]) $\boldsymbol{\mathcal{R}_1}$, $\boldsymbol{\mathcal{R}_m}$, $\boldsymbol{\mathcal{R}_{btt}}$, $\boldsymbol{\mathcal{R}_{tt}}$, $\boldsymbol{\mathcal{R}_{wtt}}$, *and* $\boldsymbol{\mathcal{R}}$ *are pairwise not elementarily equivalent.*

Proof. $\boldsymbol{\mathcal{R}_1}$ differs from all the following structures because it is not an uppersemilattice (X.7.4), while all the others are.

$\boldsymbol{\mathcal{R}_m}$ differs from all the following structures because there are no nontrivial r.e. m-degrees joining to $\mathbf{0}'_m$ (see X.5.8 and the comments after X.6.14).

$\boldsymbol{\mathcal{R}_{btt}}$ differs from $\boldsymbol{\mathcal{R}_{tt}}$ because the minimal r.e. btt-degrees are not bounded below $\mathbf{0}'_{btt}$ (X.7.10 and X.7.18). It differs from $\boldsymbol{\mathcal{R}_{wtt}}$ and $\boldsymbol{\mathcal{R}}$ because there are minimal r.e. btt-degrees (X.7.8 and X.2.8, which applies to wtt-degrees as well as Turing degrees, because permitting preserves wtt-reducibility by p. I.338).

$\boldsymbol{\mathcal{R}_{wtt}}$ differs from $\boldsymbol{\mathcal{R}}$ because it is distributive (X.6.27.a and X.6.27.c). $\square$

Bulitko [1980] and Selivanov [1982] have shown that the only possible truth-table like reducibilities on r.e. sets are the following: $\leq_m$, $\leq_l$, $\leq_d$, $\leq_c$, $\leq_p$, and $\leq_{tt}$ (see pp. I.268 and I.331 for definitions). Together with $\leq_1$, $\leq_{wtt}$, $\leq_Q$ and $\leq_T$, we thus have 10 distinct reducibilities on r.e. sets. The proof of the result above, the following exercises, and the existence of minimal r.e. degrees for all truth-table reducibilities leave only 5 cases open for elementary inequivalence, among the 45 possible cases. These have been settled by Degtev [1979], [1982], [1983], [1983a], Omanadze [1984] and Ambos-Spies (the latter two have proved that the Splitting Theorem fails for $\boldsymbol{\mathcal{R}_Q}$). Thus *the 10 reducibilities are all elementarily inequivalent on the r.e. sets.*

Exercises X.7.23 a) $\boldsymbol{\mathcal{R}_d}$ *is not elementarily equivalent to* $\boldsymbol{\mathcal{R}_m}$ *and* $\boldsymbol{\mathcal{R}_{btt}}$. (Marchenkov [1976a]) (Hint: by the comments after X.6.14, there are r.e. d-degrees joining to $\mathbf{0}'_d$. Also, as in X.7.18, the minimal r.e. d-degrees are bounded below $\mathbf{0}'_d$.)

	$\mathcal{R}_m$	$\mathcal{R}_{btt}$	$\mathcal{R}_{tt}$	$\mathcal{R}_{wtt}$	$\mathcal{R}$
Minimal degrees	yes X.5.11	yes X.7.8	yes X.7.16	no X.2.8.1	no X.2.8.1
Minimal degrees bounded	no X.5.21	no X.7.10	yes X.7.18	– –	– –
Density Theorem	no X.5.11	no X.7.8	no X.7.16	yes X.6.3	yes X.6.3
$\mathbf{0}$ branches	yes X.5.4	yes X.5.4	yes X.5.4	yes X.6.5	yes X.6.5
Capping Theorem	yes X.5.22	yes X.7.11	no X.7.19	no X.6.24	no X.6.24
Branching Theorem	yes X.5.5	?	yes X.7.14	yes X.6.8.a	no X.6.8.c
$\mathbf{0}'$ splits	no X.5.9	yes p. 553	yes p. 553	yes p. 553	yes X.6.13
Cupping Theorem	no X.5.8	no p. 592	no p. 598	no X.6.17.a	no p. 556
Splitting Theorem	no X.5.9	no X.7.8	no X.7.16	yes X.6.13	yes X.6.13
Diamond Theorem	no X.5.9	yes X.7.12.a	yes X.7.20.a	no X.6.23	no X.6.23
Distributivity	yes VI.1.8	no X.7.12.b	no X.7.20.b	yes X.6.27.c	no X.6.27.a

Figure X.1: A comparison of degree theories

b) *$\mathcal{R}_p$ is not elementarily equivalent to $\mathcal{R}_m$ and $\mathcal{R}_{btt}$.* (Marchenkov [1976a]) (Hint: as for part a).)

c) *$\mathcal{R}_c$ is not elementarily equivalent to $\mathcal{R}_d$, $\mathcal{R}_p$, $\mathcal{R}_{btt}$, and $\mathcal{R}_{tt}$.* (Degtev [1979]) (Hint: we show that there are no nontrivial r.e. c-degrees joining to $\mathbf{0}'_c$. If $\mathcal{K} \leq_c A \oplus B$ then

$$x \in \mathcal{K} \Leftrightarrow D_{f(x)} \subseteq A \wedge D_{g(x)} \subseteq B \Leftrightarrow \langle f(x), g(x) \rangle \in A^\omega \cdot B^\omega,$$

where $C^\omega = \{z : D_z \subseteq C\}$. So $\mathcal{K} \leq_m A^\omega \cdot B^\omega$ and, as A^ω and B^ω are r.e., $\mathcal{K} \leq_m A^\omega$ or $\mathcal{K} \leq_m B^\omega$ by III.9.3. Thus $\mathcal{K} \leq_c A$ or $\mathcal{K} \leq_c B$.)

d) *$\mathcal{R}_Q$ is not elementarily equivalent to $\mathcal{R}_1$, $\mathcal{R}_m$, $\mathcal{R}_{btt}$, $\mathcal{R}_{tt}$, and $\mathcal{R}_{wtt}$.* (Degtev, Ambos-Spies) (Hint: if A and B are disjoint r.e. sets, then $A, B \leq_Q A \cup B$. Thus, by the Sacks Splitting Theorem, every nonzero r.e. Q-degree bounds incomparable r.e. Q- degrees, and is not minimal. This leaves open the case of wtt-degrees, which is taken care of by the fact that the r.e. Q-degrees are not distributive, by the same proof as in X.6.27.a.)

To conclude our comparative study of r.e. degrees theories, we summarize in the figure on the opposite page the most important structural results that we have proved or discussed.

X.8 Structure Inside Degrees ⋆

In the last section we have compared various notions of degree from the point of view of the structure they induce on the r.e. sets. Despite the elementary differences proved in X.7.22, we have noted a resemblance of the methods of proofs in the study of truth-table reducibilities on the one hand, and of weak truth-table and Turing reducibilities on the other.

In the present section we take a different perspective and analyze the possible structure of r.e. degrees of one type inside r.e. degrees of another, again discovering the same dichotomy. More precisely, we prove that a collapse can occur between 1-degrees and m-degrees, m-degrees and tt-degrees, wtt-degrees and T-degrees but not between tt-degrees and wtt-degrees.

Inside many-one degrees

Recall (VI.6.8) that an m-degree is called **irreducible** if it consists of only one 1-degree. There is no ambiguity in talking of irreducible r.e. m-degrees, since an r.e. m-degree contains only r.e. sets. In particular, the following results proved for m-degrees transfer to r.e. m-degrees.

Proposition X.8.1 (Young [1966a]) *An r.e. m-degree contains either only one or infinitely many r.e. 1-degrees.*

Proof. See VI.6.13. $\Box$

We also know that both cases can occur, since $\mathbf{0}_m$ contains infinitely many 1-degrees (p. I.321) and $\mathbf{0}'_m$ contains only one 1-degree (III.7.5).

More generally, *every nonrecursive r.e. T-degree contains an irreducible r.e. m-degree*, because it contains a simple, nonhypersimple set A (III.3.18), and the m-degree of A^ω is irreducible (see X.8.2.c).

In the opposite direction, *every nonrecursive r.e. T-degree also contains a nonirreducible r.e. m-degree*, because it contains a simple set A (III.2.14), whose cylindrification (VI.6.1) $A \cdot N$ is obviously not simple, so $A \equiv_m A \cdot N$ (VI.6.2) but $A \not\equiv_1 A \cdot N$.

Exercises X.8.2 Perfect r.e. sets. Recall (VI.6.12) that a set is **perfect** if it is η-closed, for some nontrivial r.e. equivalence relation η whose only recursive η-closed sets are $\emptyset$ and ω; and that a perfect set has irreducible m-degree. In the following exercises we let $A^\omega = \{x : D_x \subseteq A\}$.

a) *A creative set is perfect.* (Ershov [1971]) (Hint: let A and B be recursively inseparable r.e. sets. If f m-reduces A to $\mathcal{K}$, then $\mathcal{K}$ and $f(B)$ are recursively inseparable. Since $\mathcal{K} \equiv_m \mathcal{K}^\omega$, it is enough to show that $\mathcal{K}^\omega$ is perfect. Consider the equivalence relation $x\eta y$ which holds in any of the following cases: D_x and D_y are both contained in $\mathcal{K}$; or they both intersect $f(B)$; or their symmetric difference is contained in $\mathcal{K}$. $\mathcal{K}^\omega$ is obviously η-closed. By considering the η-closed set $\{x : D_x \cap \mathcal{K} \neq \emptyset\}$ and the two cases of whether it is contained in R or in $\overline{R}$, it follows that no nontrivial recursive η-closed set R exists.)

b) *A simple set is not perfect.* (Dekker and Myhill [1960]) (Hint: its m-degree is not irreducible, see VI.6.14.b.)

c) *If A is simple and not hypersimple, A^ω is perfect.* (Jockusch [1969], Ershov [1971]) (Hint: let

$$x \in B \;\Leftrightarrow\; (\exists n)(D_{f(n)} - D_x \subseteq A),$$

where $\{D_{f(n)}\}_{n\in\omega}$ witnesses the nonhypersimplicity of A. Then A^ω and B are disjoint, and nonseparable by sets recursive and closed under the following equivalence relation: $x\eta y$ if and only if the symmetric difference of D_x and D_y is contained in A. Modify the proof of part a) to show that $(A^\omega)^\omega$, which is recursively isomorphic to A^ω, is perfect.)

d) *If A is hypersimple, then A^ω is not perfect.* (Denisov [1974]) (Hint: see VI.6.12.b, since $A^\omega \leq_{tt} A$.)

Degtev [1973] has proved that *not every irreducible r.e. m-degree contains a perfect set*.

Exercises X.8.3 Structure of 1-degrees inside a nonirreducible r.e. m-degree.

a) *There is an infinite chain.* (Young [1966a]) (Hint: see the proof of VI.6.13.)

b) *There is an infinite antichain.* (Degtev [1973]) (Hint: given an r.e. set A which is not a cylinder, we show how to build r.e. sets B and C such that $A \equiv_m B \equiv_m C$, and B and C are 1-incomparable. The full result only requires notational changes. We build r.e. disjoint boxes B_x and let $B = \bigcup_{x \in A} B_x$. To obtain $A \leq_m B$, we want to code A into B:

$$x \in A \ \Leftrightarrow\ 2x \in B.$$

We thus start from $B_{x,0} = \{2x\}$. To obtain $B \leq_m A$, it is enough to make sure that $\bigcup_{x\in\omega} B_x = \omega$. Then, if $h(x)$ is the unique n such that $x \in B_n$,

$$x \in B \ \Leftrightarrow\ B_{h(x)} \subseteq B \ \Leftrightarrow\ h(x) \in A.$$

We have similar boxes C_x for C. To have $B \not\leq_1 C$, make sure that in the opposite case almost every B_x is infinite but each C_x is finite, by putting a new element in B_x whenever φ_e has a new length of agreement at stage s, for $e \leq x \leq s$. If $B \leq_1 C$, then A is a cylinder by VI.6.3, since we can let

$$\mathcal{W}_{g(x)} = \{n : \varphi_e(z) \in C_n \text{ for some } z \in B_x\}.$$

$\mathcal{W}_{g(x)}$ is infinite because B_x is infinite, each C_n is finite, and φ_e is one-one. In addition,

$$x \in A \ \Rightarrow\ B_x \subseteq B \ \Rightarrow\ \mathcal{W}_{g(x)} \subseteq A \qquad \text{and} \qquad x \in \overline{A} \ \Rightarrow\ \mathcal{W}_{g(x)} \subseteq \overline{A}.)$$

c) *There is a greatest element.* (Rogers [1967]) (Hint: if A is in the given m-degree, consider $A \cdot N$.)

d) *There is no maximal element.* (Young [1966a]) (Hint: see the proof of VI.6.13.)

e) *There may be no least element.* (Dekker and Myhill [1960]) (Hint: see X.8.2.b.)

f) *If there is a minimal element, there is a least element.* (Degtev [1976a]) (Hint: let A and B be 1-incomparable r.e. sets of the same m-degree. If f m-reduces A to B, let

$$x \in R \ \Leftrightarrow\ (\exists y < x)(f(x) = f(y)).$$

R is recursive, and if $C = A \cup R$, then $C \leq_1 A, B$ and $C \equiv_m A$. Since A and B are 1-incomparable, $C <_1 A, B$. So if A has 1-degree minimal among those in its m-degree, C has least 1-degree.)

g) *There may be a least element.* (Hint: by part f), considering e.g. the m-degree of an r.e. set of minimal 1-degree.)

Degtev [1976a] shows that *there can be exactly n minimal elements, for any given n*. It is not known if there can be infinitely many minimal elements. Degtev also shows that *the structure of 1-degrees inside a given r.e. m-degree can be dense.*

Inside truth-table degrees

We now look at r.e. m-degrees inside r.e. tt-degrees. Since in VI.6.15 we have proved that every nonzero tt-degree contains infinitely many m-degrees, the following is the only possible collapse.

Definition X.8.4 (Degtev [1973]) *An r.e. set A has* **singular *tt*-degree** *if*

$$(\forall B \ r.e.)(A \equiv_{tt} B \ \Rightarrow\ A \equiv_m B).$$

Theorem X.8.5 (Degtev [1973]) *There exists a nonzero singular r.e. tt-degree.*

Proof. We want to build a coinfinite r.e. set A such that

$$\begin{array}{lcl} P_e & : & \mathcal{W}_e \text{ infinite } \Rightarrow \mathcal{W}_e \cap A \neq \emptyset \\ N_{e,i,j} & : & (\mathcal{W}_e \leq_{tt} A \text{ via } \varphi_i) \wedge (A \leq_{tt} \mathcal{W}_e \text{ via } \varphi_j) \Rightarrow \mathcal{W}_e \equiv_m A. \end{array}$$

To satisfy P_e we proceed as usual, by waiting until an element $x \geq 2e$ not restrained by negative requirements of higher priority appears in $\mathcal{W}_e$, and then put it into A.

We can thus concentrate on the strategy to satisfy $N_{e,i,j}$. First, we consider the subrequirement

$$(\mathcal{W}_e \leq_{tt} A \text{ via } \varphi_i) \Rightarrow \mathcal{W}_e \leq_m A.$$

Any m-complete set obviously satisfies all these subrequirements (although not the full requirements, since not every tt-complete set is m-complete), but here we show how to construct A directly.

To build a recursive function f_e such that

$$x \in \mathcal{W}_e \Leftrightarrow f_e(x) \in A,$$

given x we wait for the first stage s such that $l(i,s) > x$, where

$$l(i,s) = \max\{z : (\forall y < z)(y \in \mathcal{W}_{e,s} \Leftrightarrow A_s \models \sigma_{\varphi_{i,s}(y)})\}.$$

If such a stage never comes, then $\mathcal{W}_e \not\leq_{tt} A$ via φ_i and there is nothing to do. Otherwise, there are various possibilities.

If $x \in \mathcal{W}_{e,s}$, it is enough to let $f_e(x)$ be a fixed element of A. Then

$$x \in \mathcal{W}_e \Leftrightarrow f_e(x) \in A$$

because both sides are true.

If $x \notin \mathcal{W}_{e,s}$, then $A_s \not\models \sigma_{\varphi_i(x)}$ by the choice of s. There are only finitely many possible extensions A^* of A_s that involve the finitely many elements of $\sigma_{\varphi_i(x)}$ not yet in A, and for each of them we can effectively determine whether $A^* \models \sigma_{\varphi_i(x)}$ or not, since $\sigma_{\varphi_i(x)}$ is a truth-table. .

If for all these extensions A^* we have $A^* \not\models \sigma_{\varphi_i(x)}$, it is enough to let $f_e(x)$ be a fixed element of $\overline{A}$ (i.e. a permanently restrained one). Then either $\mathcal{W}_e \not\leq_{tt} A$ via φ_i, or

$$x \in \mathcal{W}_e \Leftrightarrow f_e(x) \in A$$

holds because both sides are false (since $A \not\models \sigma_{\varphi_i(x)}$, and hence $x \notin \mathcal{W}_e$).

If instead there are extensions A^* of A such that $A^* \models \sigma_{\varphi_i(x)}$, then the possibility exists that $x \in \mathcal{W}_e$, and hence $A \models \sigma_{\varphi_i(x)}$. If this happens, then A will change from A_s to some A^* as above, and hence some element of $\sigma_{\varphi_i(x)}$ that is not in A_s will enter A. We want to choose one such element as $f_e(x)$, but the choice is complicated by the fact that there are in general many possible extensions A^* of A_s, and we do not know in advance which one will be realized. If we pick $f_e(x)$ in the wrong A^*, then x could enter $\mathcal{W}_e$ without $f_e(x)$ entering A, and f_e would not be an m-reduction of $\mathcal{W}_e$ to A.

We thus build A in such a way that if $\overline{A}_s = \{a_0^s < a_1^s < \cdots\}$, then whenever we put in A at stage $s+1$ some a_n^s, we also put every a_m^s with $n \leq m \leq s$ (this makes A semirecursive, as in the proof of III.5.19). The possible extensions A^* of A_s as above thus reduce to ones of the form

$$A_n^* = A_s \cup \{a_m^s : n \leq m \leq s\}.$$

This simplifies the choice of $f_e(x)$, since we only have to look for the unique a_n^s with $n \leq s$ that provides a cutting point, i.e. such that $A_m^* \models \sigma_{\varphi_i(x)}$ if $m \leq n$, and $A_m^* \not\models \sigma_{\varphi_i(x)}$ if $m > n$.

If such an n exists, we can let $f_e(x) = a_n^s$, and argue that if $\mathcal{W}_e \leq_{tt} A$ via φ_i then

$$x \in \mathcal{W}_e \ \Leftrightarrow\ f_e(x) \in A.$$

In one direction, if $x \in \mathcal{W}_e$, then $A \models \sigma_{\varphi_i(x)}$, i.e. $A = A_m^*$ on the elements of $\sigma_{\varphi_i(x)}$, for some $m \leq n$. In particular, $f_e(x) = a_n^s \in A$. In the opposite direction, if $x \notin \mathcal{W}_e$, then $A \not\models \sigma_{\varphi_i(x)}$, i.e. $A = A_m^*$ on the elements of $\sigma_{\varphi_i(x)}$, for some $m > n$. In particular, $f_e(x) = a_n^s \notin A$.

Is such an n does not exist, then there is an n such that $A_n^* \not\models \sigma_{\varphi_i(x)}$ and $A_{n+1}^* \models \sigma_{\varphi_i(x)}$, and we can force $\mathcal{W}_e \not\leq_{tt} A$ via φ_i as follows:

1. First we put a_{n+1}^s in A, so that $A_{s+1} \models \sigma_{\varphi_i(x)}$, and restrain a_n^s, so that this cannot change.

2. Then we wait for a stage $t > s$ such that $l(i,t) > x$. If there is no such stage, then $\mathcal{W}_e \not\leq_{tt} A$ via φ_i.

3. Otherwise, by the choice of t,

$$x \in \mathcal{W}_{e,t} \ \Leftrightarrow\ A_t \models \sigma_{\varphi_i(x)}.$$

   Since $A_{s+1} \models \sigma_{\varphi_i(x)}$, and a_n^s was restrained, we still have $A_t \models \sigma_{\varphi_i(x)}$. Thus $x \in \mathcal{W}_{e,t}$ and hence $x \in \mathcal{W}_e$. We now put a_n^s in A, thus making $A_{t+1} \not\models \sigma_{\varphi_i(x)}$, and restrain a_{n-1}^s, so that $A \not\models \sigma_{\varphi_i(x)}$. This ensures that $\mathcal{W}_e \not\leq_{tt} A$ via φ_i.

We would now like to satisfy the symmetric subrequirement

$$(A \leq_{tt} \mathcal{W}_e \text{ via } \varphi_j) \;\Rightarrow\; A \leq_m \mathcal{W}_e$$

in a similar way, by building a recursive function g_e such that

$$x \in A \;\Leftrightarrow\; g_e(x) \in \mathcal{W}_e.$$

We cannot proceed directly as above, because $\mathcal{W}_e$ is given and not built, but there is no need to satisfy the two subrequirements separately, since their conjunction is stronger than $N_{e,i,j}$. We thus go back to the original requirement $N_{e,i,j}$, and notice that if really $\mathcal{W}_e \leq_{tt} A$ via φ_i, then we can control $\mathcal{W}_e$ indirectly through A.

To define $g_e(x)$, we thus wait for the first stage s such that $l_1(j,s) > x$, where

$$l_1(j,s) = \max\{z : (\forall y < z)[(y \in A_s \;\Leftrightarrow\; \mathcal{W}_{e,s} \models \sigma_{\varphi_j(y)}) \wedge \\ (l(i,s) > \max \sigma_{\varphi_j(y)})]\}.$$

and proceed as above, with the only difference that we do not look at the extensions A_n^* of A_s, but rather at the extensions $\mathcal{W}_n^*$ of $\mathcal{W}_{e,s}$ forced by A_n^* through the tt-reduction of $\mathcal{W}_e$ to A via φ_i.

We now proceed as above, by looking for a cutting point n of the tt-reduction of A to $\mathcal{W}_e$ via φ_j, defining $g_e(x)$ to be any element of $\mathcal{W}_n^* - \mathcal{W}_{n+1}^*$ if such an n exists, and forcing $A \not\leq_{tt} \mathcal{W}_e$ via φ_j otherwise. However, the proof that f_e is an m-reduction used in a crucial way the fact, ensured by construction, that $A_m^* \supseteq A_n^*$ if $m \leq n \leq s$. If we want to argue in a similar way that g_e is an m-reduction, we must then have $\mathcal{W}_m^* \supseteq \mathcal{W}_n^*$ if $m \leq n \leq s$.

If the condition is satisfied, we proceed as above. If not, then $\mathcal{W}_{n+1}^* \not\subseteq \mathcal{W}_n^*$ for some $n < s$, and we can force $A \not\equiv_{tt} \mathcal{W}_e$ via (φ_i, φ_j) as follows:

1. First we put a_{n+1}^s in A, and restrain a_n^s.
2. Then we wait for a stage $t > s$ such that $l_1(j,t) > a_n^s$. If there is no such stage, then $A \not\equiv_{tt} \mathcal{W}_e$ via (φ_i, φ_j).
3. Otherwise, by the choice of t, $\mathcal{W}_{e,t} = \mathcal{W}_{n+1}^*$ on the elements of $\sigma_{\varphi_j(x)}$. We now put a_n^s in A, and restrain a_{n-1}^s. This ensures that $A \not\equiv_{tt} \mathcal{W}_e$ via (φ_i, φ_j), otherwise at the end $\mathcal{W}_e = \mathcal{W}_n^*$ on the elements of $\sigma_{\varphi_j(x)}$, which is impossible because it would require taking some elements of $\mathcal{W}_{n+1}^*$ out of $\mathcal{W}_e$.

The strategies for all requirements can now be put together as in a typical finite injury priority argument. $\square$

Degtev [1973] originally proved the result indirectly, by using rigid semirecursive η-maximal sets. Downey [1988] gave the direct proof above, and building on it proved a number of results on the distribution of singular r.e. *tt*-degrees. In particular, *not every r.e. T-degree contains a singular r.e. tt-degree, but the r.e. T-degrees that do are dense*. Moreover, $\mathbf{0}'$ *contains a singular r.e. tt-degree*.

Downey [1989c] and Cholak and Downey [1994] have proved that, *for every* $n \geq 1$*, there exist r.e. tt-degrees with exactly* n *r.e. m-degrees*. In particular, the analogue of X.8.1 fails.

Proposition X.8.6 (Fischer [1963]) $\mathbf{0}'_{tt}$ *contains infinitely many r.e. m-degrees.*

Proof. Let A be the simple, *tt*-complete set of III.3.5. There is a strong array $\{F_x\}_{x\in\omega}$ such that

$$x \in \mathcal{K} \Leftrightarrow F_x \subseteq A.$$

We know that, in general, $A \equiv_{tt} A \cdot A$ and $A \leq_m A \cdot A$. Suppose $A \cdot A \leq_m A$. Then

$$x \in A \wedge y \in A \Leftrightarrow g(x,y) \in A$$

for some recursive function g, and it is easy to define a recursive function h such that

$$x \in \mathcal{K} \Leftrightarrow h(x) \in A,$$

contradicting the fact that a simple set is not m-complete. Thus $A <_m A \cdot A$.

More generally, consider the sequence $A_0 \leq_m A_1 \leq_m \cdots$, where

$$A_0 = A \qquad \text{and} \qquad A_{n+1} = A_n \cdot A_n.$$

If $A_{n+1} \leq_m A_n$ for some n, then $\mathcal{K} \leq_m A_n$ as above. But $A_n \leq_{btt} A$ by definition, hence $\mathcal{K} \leq_{btt} A$, contradicting the fact that a simple set is not *btt*-complete (III.8.8). □

Exercises X.8.7 The structure of *tt*-complete sets.

a) *The complete btt-degree contains infinitely many r.e. m-degrees.* (Kallibekov [1973], Kobzev [1974]) (Hint: we build r.e. sets A_n such that $\mathcal{K} \leq_{btt} A_n$, and $A_m \not\leq_m A_n$ whenever $m \neq n$.

To have $\mathcal{K} \leq_{btt} A_n$, we let

$$x \in \mathcal{K} \Leftrightarrow \{2x, 2x+1\} \cap A_n \neq \emptyset.$$

If at stage $s+1$ we have $x \in \mathcal{K}_s$ but $\{2x, 2x+1\} \cap A_{n,s} = \emptyset$, we put in A_n the smallest between $2x$ and $2x+1$ that is not restrained. If both are restrained, we put in A_n the one restrained by a condition of lower priority.

To have $A_m \not\leq_m A_n$ via φ_e, note that

$$\{2x, 2x+1\} \cap A_m \neq \emptyset \;\Rightarrow\; x \in \mathcal{K}$$

by the first part of the construction. Choose disjoint recursive subsets $R_{m,n,e}$ of $\mathcal{K}$. We only look at $2a_0, 2a_1, \ldots$, where $a_0, a_1, \ldots$ is an enumeration of $R_{m,n,e}$ in order of magnitude. Suppose $2a_i$ is the current witness at stage $s+1$, $\varphi_{e,s}(2a_i)$ converges and $2a_i \notin A_{m,s}$. If $\varphi_e(2a_i) \notin A_{n,s}$, put $2a_i$ into A_m and restrain $\varphi_e(2a_i)$ from entering A_n. If $\varphi_e(2a_i) \in A_{n,s}$, restrain $2a_i$ from entering A_m. The strategy succeeds because only one element is restrained to satisfy the second type of requirement, but there are two to satisfy the first type.

Alternatively, one can use upward density X.5.2, and the fact that there is a *btt*-complete, m-incomplete set, e.g. $A \oplus B$ where A and B are a Sacks splitting of $\mathcal{K}$. Then $A \oplus B \equiv_{btt} \mathcal{K}$, but by X.5.8 $A \oplus B$ cannot be m-complete, otherwise so would be one of A or B.)

b) *The complete tt-degree contains infinitely many r.e. btt-degrees.* (Kallibekov [1973]) (Hint: we build r.e. sets A_n such that $\mathcal{K} \leq_{tt} A_n$, and such that $A_m \not\leq_{btt} A_n$ whenever $m \neq n$.

To have $\mathcal{K} \leq_{tt} A_n$, we let

$$x \in \mathcal{K} \;\Leftrightarrow\; I_x \cap A_n \neq \emptyset,$$

where $I_0 = \{0\}$, $I_1 = \{1, 2\}$, ... and I_x has $x+1$ elements. To spoil *btt*-reductions with norm n, we have to take care of n elements and, for almost every x, I_x has more than n elements. See also p. I.343. Alternatively, one can use upward density X.5.2, and the fact that there is a *tt*-complete, *btt*-incomplete set, e.g. a simple *tt*-complete set.)

Lachlan [1965], Bulitko [1980] and Degtev [1981], [1982], [1983] have studied the structure of *tt*-complete sets w.r.t. many reducibilities caught in between $\leq_m$ and $\leq_{tt}$. Stephan [199?] has proved that the structure of r.e. *btt*-degrees inside $\mathbf{0}'_{tt}$ is complicated, in particular it embeds the lattice $\boldsymbol{\mathcal{E}}$ of the r.e. sets. Kobzev [1977] has proved that $\mathbf{0}'_{tt}$ contains infinitely many r.e. *bwtt*-degrees (recall that *btt*-completeness and *bwtt*-completeness coincide, see p. I.347).

A nonsingular r.e. *tt*-degree can contain a *greatest r.e. m-degree* (e.g. $\mathbf{0}'_{tt}$ contains the m-degree of $\mathcal{K}$). Downey [1988a] has proved that also the opposite case can happen (although, obviously, not for an r.e. *tt*-degree containing only finitely many r.e. m-degrees).

Cholak and Downey [1994] have proved that any finite distributive lattice can be realized as the structure of r.e. m-degrees inside an r.e. *tt*-degree (since such a structure is distributive if it is a lattice, this completely characterizes the finite lattices that can occur, but Downey [1989c] has proved that the structure is not always a lattice). In particular, a nonsingular r.e. *tt*-degree can contain a *least r.e. m-degree*. Downey [1989c] has proved that also the opposite case can happen.

Inside weak truth-table degrees

We now look at r.e. *tt*-degrees inside r.e. *wtt*-degrees. This case differs from all the others treated in this section, because there is no collapse. This is proved by an extension of the proof of III.9.1, which showed that *tt*-completeness and *wtt*-completeness do not coincide.

Proposition X.8.8 (Lachlan [1975], Cohen [1975], Kobzev [1976]) *Every nonzero r.e. wtt-degree contains infinitely many r.e. tt-degrees.*

Proof. We refer to the proof of III.9.1. Given a nonrecursive r.e. set B, we build r.e. sets A_n such that $A_n \equiv_{wtt} B$, and $A_m \not\leq_{tt} A_n$ whenever $m \neq n$. We define a double sequence of boxes $I_{n,x}$ and equivalence relations η_n for A_n.

- To obtain $B \leq_{wtt} A_n$, we ensure that
$$x \in B_{s+1} - B_s \;\Rightarrow\; I_{n,x} \cap (A_{n,s+1} - A_{n,s}) \neq \emptyset.$$
- To obtain $A_n \leq_{wtt} B$, at any stage $s+1$ such that $x \in B_{s+1} - B_s$ we can take action for
$$R_{m,n,e} \;:\; \neg(A_m \leq_{tt} A_n \text{ via } \varphi_e)$$
only on the boxes $I_{m,y}$ or $I_{n,y}$ with $y \geq x$. Thus, for example, $I_{m,y}$ can change only when some $x \leq y$ is generated in B (for this or the previous part), and $A_n \leq_{wtt} B$.

This produces some changes in the construction of III.9.1. If $x \in B_{s+1} - B_s$, at step $s+1$ we look for the smallest $\langle m,n,e\rangle \leq x$ ($m \neq n$) such that $R_{m,n,e}$ has not yet been satisfied and no longer injured, and $\varphi_{e,s}$ converges on all elements of $I_{m,x}$. If such a number exists, we take the smallest $z \in I_{m,x} \cap \overline{A}_{m,s}$ as witness for $R_{m,n,e}$, and consider the smallest $y \geq x$ such that all elements used in $\sigma_{\varphi_e(z)}$ are in $I_{n,0} \cup \cdots \cup I_{n,y}$. Then we proceed as in III.9.1 on the boxes $I_{n,x}, \ldots, I_{n,y}$, thus ensuring
$$z \notin A_m \;\Leftrightarrow\; A_n \models \sigma_{\varphi_e(z)},$$
unless some change happens in $I_{n,0}, \ldots, I_{n,x-1}$. Finally, we also ensure that, for each n, $I_{n,x} \cap (A_{n,s+1} - A_{n,s}) \neq \emptyset$.

To show that $R_{m,n,e}$ is satisfied when φ_e is total, we need to know that there are infinitely many stages $s+1$ in which, if $x \in B_{s+1} - B_s$, $\varphi_{e,s}$ is defined on all elements of $I_{m,x}$ and A_n has settled on $I_{n,0}, \ldots, I_{n,x-1}$. Since a change can occur on these boxes only if some element less that x enters B at some later stage, it is enough (to ensure that A_n has settled on $I_{n,0}, \ldots, I_{n,x-1}$) to know that B has settled up to x. Let
$$h(x) = \mu s.\, (\forall z)(z \in I_{m,x} \;\Rightarrow\; \varphi_{e,s}(z)\!\downarrow).$$

h is total because φ_e is. Since B is nonrecursive, there are infinitely many elements x generated at stages $s+1$ such that $B_s[x] = B[x]$ and $s \geq h(x)$ (otherwise to compute $B[x]$ it is enough, with at most finitely many exceptions, to compute $B_{h(x)}[x]$). But then, as soon as $R_{m,n,e}$ has highest priority and one of those x is generated, $R_{m,n,e}$ is permanently satisfied. $\Box$

The proof shows that *every nonzero r.e. wtt-degree contains an infinite antichain of r.e. tt-degrees.*

Exercise X.8.9 *Every nonzero r.e. wtt-degree contains an infinite chain of r.e. tt-degrees.* (Kobzev [1976]) (Hint: modify the proof above to ensure that, for every n, $A_n \not\leq_{tt} A_0 \oplus \cdots \oplus A_{n-1}$.)

An r.e. *wtt*-degree can contain a *greatest r.e. tt-degree* (e.g. $\mathbf{0}'_{wtt}$ contains the *tt*-degree of $\mathcal{K}$). Downey and Jockusch [1987] have proved that the opposite case can also occur.

An r.e. *wtt*-degree can contain a *least r.e. tt-degree* (if A is a nonrecursive, semirecursive η-maximal set and $A \equiv_T B$, then $A \leq_{tt} B$ by X.7.15). Downey has proved that the opposite case can also occur.

Inside Turing degrees

We now look at r.e. *wtt*-degrees inside r.e. T-degrees.

Definition X.8.10 (Ladner [1973a], Downey [1987b]) *An r.e. set A has* **contiguous degree** *if*

$$(\forall B \text{ r.e.})(A \equiv_T B \ \Rightarrow\ A \equiv_{wtt} B).$$

A has **strongly contiguous degree** *if*

$$(\forall B)(A \equiv_T B \ \Rightarrow\ A \equiv_{wtt} B).$$

Exercise X.8.11 *If the degree of A is contiguous and $B \leq_T A$, then $B \leq_{wtt} A$.* (Hint: if $A \oplus B \equiv_T A$ then $B \leq_{wtt} A \oplus B \equiv_{wtt} A$.)

Proposition X.8.12 (Ladner and Sasso [1975]) *An r.e. Turing degree contains either only one or infinitely many r.e. wtt-degrees.*

Proof. If a given r.e. Turing degree contains two distinct r.e. *wtt*-degrees, it also contains two distinct comparable r.e. *wtt*-degrees (by taking the l.u.b. of the two original ones). Then one can apply density of the r.e. *wtt*-degrees. $\Box$

We now show that both possibilities may occur.

Proposition X.8.13 (Friedberg and Rogers [1959]) $\mathbf{0}'$ *is not contiguous.*

Proof. By III.3.13, there is a T-complete hypersimple set. By III.8.16, it cannot be *wtt*-complete. □

A different proof of the previous result follows from X.8.15.a.

Theorem X.8.14 (Ladner [1973a], Downey [1987b]) *There exists a nonzero strongly contiguous degree.*

Proof. We want to build an r.e. set A such that

$$\begin{array}{lcl} P_e & : & \overline{A} \neq \mathcal{W}_e \\ N_{e,i} & : & A \simeq \{e\}^B \wedge B \simeq \{i\}^A \Rightarrow A \equiv_{wtt} B. \end{array}$$

To satisfy P_e we proceed as usual, by choosing an element z_e not yet in A and putting it into A if and when it enters $\mathcal{W}_e$. Then either $z_e \in \overline{A} \cap \overline{\mathcal{W}_e}$ or $z_e \in A \cap \mathcal{W}_e$, i.e. $\overline{A} \neq \mathcal{W}_e$.

We can thus concentrate on the strategy for $N_{e,i}$. We actually satisfy the requirement in the stronger form

$$A \simeq \{e\}^B \wedge B \simeq \{i\}^A \Rightarrow A \leq_{tt} B \wedge B \leq_{wtt} A.$$

Notice that it is impossible in general to upgrade also the remaining *wtt*-reduction to a *tt*-reduction, since no r.e. T-degree contains only one (r.e.) *tt*-degree by X.8.8.

First, we consider the subrequirement

$$A \simeq \{e\}^B \wedge B \simeq \{i\}^A \Rightarrow A \leq_{tt} B.$$

Any r.e. set having the least *tt*-degree in its own T-degree, e.g. any r.e. set of minimal *tt*-degree, would obviously satisfy all these requirements (although not the full requirements, since not every r.e. T-degree having a least *tt*-degree is contiguous by X.8.17.e), but here we show how to construct A directly.

To build a recursive function $f_{e,i}$ such that

$$x \in A \Leftrightarrow B \models \sigma_{f_{e,i}(x)},$$

given x we wait for the first stage s such that $l(e,i,s) > x$, where

$$\begin{array}{lcl} l(e,i,s) & = & \max\{z : (\forall y < z)(A_s(y) \simeq \{i\}_s^{B_s}(y))\}, \\ & & \text{where } B_s = \{e\}_s^{A_s}. \end{array}$$

If such a stage never comes, then the hypotheses of $N_{e,i}$ are not satisfied and there is nothing to do. Otherwise, there are various possibilities.

If $x \in A_s$, it is enough to let

$$\sigma_{f_{e,i}(x)} \quad = \quad b \in X,$$

where b is a fixed element if B. Then

$$x \in A \;\Leftrightarrow\; B \models \sigma_{f_{e,i}(x)}$$

because both sides are true.

If $x \notin A_s$, we try to have

$$x \in A \quad \Leftrightarrow \quad B_s[m_x] \neq B[m_x], \tag{X.5}$$

where m_x is the maximum element used in any of the computations $\{i\}_s^{B_s}(y)$, for $y \leq x$. Then it is enough to let

$$\sigma_{f_{e,i}(x)} \quad = \quad (\bigvee_{z \leq m_x \,\wedge\, z \notin B_s} z \in X) \;\vee\; (\bigvee_{z \in B_s} z \notin X),$$

to have

$$x \in A \;\Leftrightarrow\; B \models \sigma_{f_{e,i}(x)}.$$

Before we proceed, we notice that if we only tried to construct a contiguous degree, as opposed to a *strongly* contiguous degree, then the second disjunction of the truth-table could be dropped (since we never take elements out of an r.e. set). The only (not required) advantage would thus be the improvement of the condition $A \leq_{tt} B$ to $A \leq_d B$.

We split condition X.5 as follows:

$$x \in A - A_s \;\Leftrightarrow\; A_s[x] \neq A[x] \;\Leftrightarrow\; B_s[m_x] \neq B[m_x].$$

We now see that the left-to-right implications are always true, while the right-to-left implications can be achieved by two complementary actions: *dumping* and *confirming*.

- $x \in A - A_s \;\Leftrightarrow\; A_s[x] \neq A[x]$

  If $x \in A - A_s$, then obviously $A_s[x] \neq A[x]$.

  If $A_s[x] \neq A[x]$, then A_s changes on some $y \leq x$. To force a change on x we use *dumping*, i.e. whenever we put an element y into A at stage $s+1$, we also put all elements x such that $y \leq x \leq s$. As in III.5.19, this makes A semirecursive (all known constructions of r.e. T-degrees with a least r.e. tt-degree use semirecursive sets).

- $A_s[x] \neq A[x] \;\Leftrightarrow\; B_s[m_x] \neq B[m_x]$

  If $A_s[x] \neq A[x]$, then A_s changes on some $y \leq x$. Since $x < l(e,i,s)$, this means that $\{i\}_s^{B_s}$ changes on some $y \leq x$, i.e. B_s changes below the use of $\{i\}_s^{B_s}(y)$, for some $y \leq x$. Thus $B_s[m_x] \neq B[m_x]$.

  If $B_s[m_x] \neq B[m_x]$, then B_s changes below the use of $\{i\}_s^{B_s}(y)$, for some $y \leq x$. Since $B_s \simeq \{e\}_s^{A_s}$, A_s must change below s. The problem is that there is no reason to suppose that A_s changes below x, i.e. it could change in the interval between x and s. Since the elements that enter A are bounded below by witnesses for positive requirements, we can avoid the problem by cancelling all the existing witnesses between x and s, thus *confirming* x at $\langle e,i\rangle$, and by putting into A only witnesses that have already been confirmed.

  When a positive requirement looses its witness at stage $s+1$, we choose a new witness for it greater than s. This makes all the new witnesses greater than any computation converging at step s, so that none of them can be injured by later action. Moreover, we choose larger witnesses for lower priority requirements, so that the cancellation procedure does not interfere with the priority order.

This takes care of the *tt*-reduction in the requirement

$$A \simeq \{e\}^B \;\wedge\; B \simeq \{i\}^A \;\Rightarrow\; A \leq_{tt} B \;\wedge\; B \leq_{wtt} A.$$

As for the remaining *wtt*-reduction, it is automatically satisfied if we restrict the action for $N_{e,i}$ to an infinite recursive set $S_{e,i}$ of stages on which the length of agreement $l(e,i,s)$ is increasing (e.g. the $\langle e,i\rangle$-maximal stages defined as in X.6.5). In particular, we confirm witnesses only at stages belonging to $S_{e,i}$.

With this modification, we can argue that $B \leq_{wtt} A$ as follows. Given x, we first find, recursively, the least stage $s_x \in S_{e,i}$ such that $l(e,i,s_x) > x$. Then we find, recursively in A, the least stage $t \in S_{e,i}$ such that $t \geq s_x$ and

$$A_t[s_x] = A[s_x].$$

Since the search for t requires the use of the oracle A only up to the recursive bound s_x, it is enough to prove

$$x \in B \;\Leftrightarrow\; x \in B_t$$

to have a *wtt*-reduction of B to A.

Suppose $B_t(x) \simeq \{e\}_t^{A_t}(x)$ is not final, i.e. A_t changes. By the choice of s_x in $I_{e,i}$, A must change above s_x. This means that some $y \geq s_x$ below the use of $\{e\}_t^{A_t}(x)$ enters A after stage t. We can suppose that y is the witness of some positive requirement (otherwise, by the dumping procedure, we can choose some smaller element that is such a witness). By the restricted confirmation procedure, y must have been chosen at some stage $v \in S_{e,i}$ such that $s_x \leq v < t$. By the dumping procedure, no number $\leq y$ has entered A after stage v (when y was chosen) but before stage t (at which y has not yet entered A), otherwise y would have entered too. In particular,

$$A_v[y] = A_t[y].$$

Since $s_x \leq y$, it follows in particular that

$$A_v[s_x] = A_t[s_x] = A[s_x],$$

contradicting the choice of t as the least such stage $\geq s_x$ with such a property.

If we only want to satisfy the negative subrequirements considered above, i.e. if we only want to build an r.e. T-degree with a least tt-degree, there is no problem in putting all strategies together as in a typical finite injury priority argument.

If we want to satisfy the full negative requirements, i.e. if we want to build a (strongly) contiguous degree, there are instead two problems. First, we do not know whether the hypotheses of a single $N_{e,i}$ are satisfied, i.e. whether $\lim_{s\to\infty} l(e,i,s)$ is infinite and the set $S_{e,i}$ exists. Second, since the construction restricts the action for the satisfaction of $N_{e,i}$ to stages in $S_{e,i}$, the various $S_{e,i}$ must be compatible among each other.

However, these problems are the same already encountered in the construction of minimal pairs, and so are the solutions. In particular, the strategies for the negative requirements can be put together by using either $\langle e, i\rangle$-maximal stages as in X.6.5, or the tree method as in X.6.6.b (see Ambos-Spies [1984] for details). $\Box$

Downey and Lempp [1997] have proved that *the class of contiguous degrees is definable in $\boldsymbol{\mathcal{R}}$*. More precisely, *a degree is contiguous if and only if the r.e. degrees below it form a distributive uppersemilattice* in the sense of VI.1.3. The condition is obviously necessary, by X.8.11 and the distributivity of $\boldsymbol{\mathcal{R}_{wtt}}$ (X.6.27.c). Since the proof that the condition is sufficient produces a strongly contiguous degree, it follows that *an r.e. degree is contiguous if and only if it is strongly contiguous*.

Ambos-Spies and Fejer [199?] have provided a different definition of the class of contiguous degrees. More precisely, *a degree is noncontiguous if and only if it is the greatest element of an embedding of N_5 in $\boldsymbol{\mathcal{R}}$* (see p. 560).

Exercises X.8.15 Contiguous degrees.

a) *Every contiguous degree is low$_2$.* (Robinson [1968], Kallibekov [1971a], Jockusch [1972b], Cohen [1975]) (Hint: If A has contiguous degree, then it is not T-complete.

By X.9.17.c, the index set $\{x : \mathcal{W}_x \leq_T A\}$ is $\Sigma_3^{0,A}$-complete. Since A has contiguous degree, by X.8.11 this index set coincides with $\{x : \mathcal{W}_x \leq_{wtt} A\}$, which is Σ_3^0. Thus $\Sigma_3^{0,A} \subseteq \Sigma_3^0$ and A is low$_2$.)

b) *Below any nonzero r.e. degree there exists a nonzero contiguous degree.* In particular, there exists a nonzero low contiguous degree. (Ladner and Sasso [1975], Downey [1987b]) (Hint: apply permitting to X.8.14.)

c) *Below any nonzero r.e. degree there exists a noncontiguous r.e. degree.* In particular, there exists a low noncontiguous degree. (Sasso [1974a]) (Hint: given an r.e. nonrecursive set B, we build r.e. sets C and A such that $C \leq_T A \leq_T B$ and $C \not\leq_{wtt} A$. By X.8.11, the degree of A is not contiguous. To obtain $A \leq_T B$ we use permitting.

To obtain $C \leq_T A$ we cannot use simple permitting, since it would give $C \leq_{wtt} A$. We thus use markers a_x^s as in X.6.8.c, such that

$$x \in C_{s+1} - C_s \ \vee\ a_x^{s+1} \neq a_x^s \ \Rightarrow\ (\exists y \leq a_x^s)(y \in A_{s+1} - A_s),$$

i.e. when a_x^s is permitted by A it can either move, or send x into C.

To obtain $C \not\leq_{wtt} A$, we spoil $C \simeq \{(e)_1\}^A$ with bound $\varphi_{(e)_2}$ by choosing a follower x and waiting for $\varphi_{(e)_2}(x)$ to converge. When a_x^s is permitted by B we put it into A, and move to an a_x^{s+1} greater than $\varphi_{(e)_2}(x)$. If and when $\{(e)_1\}^A(x) \simeq 0$, we restrain the numbers less than $\varphi_{(e)_2}(x)$ from entering A. Finally, if a_x^{s+1} becomes permitted by B we put it into A, and put x into C.)

Ambos-Spies and Fejer [1988] show that part a) cannot be improved, because *there are contiguous degrees which are not low.*

Exercise X.8.16 Contiguous *wtt*-degrees. *The contiguous wtt-degrees are bounded below* $\mathbf{0}'_{wtt}$. (Nies) (Hint: as in X.7.18, since every contiguous *wtt*-degree contains an r.e. coretraceable set, because by II.6.16 so does every r.e. T-degree.)

It is implicit in Ambos-Spies [1984] that, given any nontrivial r.e. T-degree, there is a contiguous T-degree incomparable with it. Thus any upper bound of the contiguous *wtt*-degrees must be T-complete.

The proof of X.8.14 shows that a contiguous degree can contain a *least tt-degree*. Downey and Lempp [1997] have proved that no contiguous degree contains a *greatest r.e. tt-degree*.

Exercises X.8.17 Structure of r.e. *wtt*-degrees inside a noncontiguous r.e. degree.

a) *There is an infinite chain.* (Ladner and Sasso [1975]) (Hint: by density of the r.e. *wtt*-degrees.)

b) *There is an infinite antichain.* (Hint: every countable partial ordering is embeddable between any two r.e. *wtt*-degrees.)

c) *There may be a greatest element.* (Hint: $\mathbf{0}'$ contains the *wtt*-degree of $\mathcal{K}$.)

d) *There may be no greatest or maximal element. Actually, this is so for every incomplete non-low*$_2$ *r.e. degree.* (Kallibekov [1971a], Jockusch [1972b], Cohen [1975])

(Hint: see X.8.15.a. If there is a maximal element, by taking joins there is the greatest element. If A has greatest *wtt*-degree, then $\{x : \mathcal{W}_x \leq_T A\} = \{x : \mathcal{W}_x \leq_{wtt} A\}$, since $\mathcal{W}_x \oplus A \equiv_T A$, so $\mathcal{W}_x \oplus A \leq_{wtt} A$.)

e) *There may be a least element.* (Downey and Jockusch [1987]) (Hint: build two r.e. sets A and C such that $A \equiv_T C$, $C \not\leq_{wtt} A$ and the T-degree of A has a least *tt*-degree. The last condition is ensured as in the proof of X.8.14. $A \leq_T C$ is ensured by permitting. $C \leq_T A$ is ensured as in the proof of IX.2.24, by leaving a trace in A whenever an element enters C. Finally, $C \not\leq_{wtt} A$ is ensured by a direct action.)

f) *There may be no least element. Actually, this is so for every non-low*$_2$ *r.e. degree, in particular for* $\mathbf{0}'$. (Downey and Stob [1986], Downey and Jockusch [1987], Downey and Shore [1995]) (Hint: suppose the *wtt*-degree of the r.e. set C is the least one inside its T-degree. Let $A \in \Sigma_3^{0,C}$. By X.9.20, there is a recursive function f such that $\mathcal{W}_{f(x)} \leq_T C$ and $x \in A \Leftrightarrow \mathcal{W}_{f(x)} \equiv_T C$. From $\mathcal{W}_{f(x)} \leq_T C$ and the fact that the *wtt*-degree of C is the least one, we have

$$x \in A \Leftrightarrow C \leq_T \mathcal{W}_{f(x)} \Leftrightarrow C \leq_{wtt} \mathcal{W}_{f(x)},$$

i.e. $A \in \Sigma_3^0$. So $\Sigma_3^{0,C} \subseteq \Sigma_3^0$, and C is low$_2$.)

Contiguous degrees can be used to translate results from r.e. wtt-degrees (where proofs are usually easier) to r.e. T-degrees. As an example, the next exercise shows how to obtain the existence of an r.e. noncupping T-degree (see X.6.17). One cannot obtain the full Non-Cupping Theorem for $\boldsymbol{\mathcal{R}}$ (i.e. that $\mathbf{0}'$ is itself noncupping) this way, because $\mathbf{0}'$ is not contiguous.

Other results obtained by using this **transfer principle** are in Stob [1983], Ambos-Spies [1984], [1985c], [1985d], Downey and Stob [1986], Downey and Welch [1986], Downey and Jockusch [1987], Downey [1987a], Ambos-Spies and Fejer [1988], Ambos-Spies and Soare [1989], Downey and Lempp [1997]. Downey [1987b] has shown that the transfer principle can be used to obtain results not only for the r.e. degrees, but also for the degrees below $\mathbf{0}'$.

Exercise X.8.18 *Every contiguous degree is noncupping.* (Ladner and Sasso [1975]) (Hint: let A be an r.e. nonrecursive set of contiguous degree. By X.6.17.b, there is an r.e. set B such that $B <_{wtt} A$, and $B \oplus C <_{wtt} A$ for every $C <_{wtt} A$. Since the degree of A is contiguous, $B <_T A$. If $C <_T A$, then $C <_{wtt} A$ by X.8.11, so $B \oplus C <_{wtt} A$ and $B \oplus C <_T A$.)

X.9 Index Sets ⋆

Index sets were introduced in II.2.8, for classes of partial recursive functions. Those are called **Kleene index sets**. In this Section we consider **Post index sets**, for classes of r.e. sets. There are genuinely Kleene index sets (i.e. index sets not recursively isomorphic to Post index sets), see Kusmina [1981] and Mohrherr [1983].

Definition X.9.1 *A set A is the* **Post index set** *of a class $\mathcal{A}$ of r.e. sets if*

$$A = \{x : \mathcal{W}_x \in \mathcal{A}\}.$$

If A is the index set of $\mathcal{A}$, we write $A = \theta\mathcal{A}$.

Exercises X.9.2 a) *Index sets are cylinders.* (Rogers [1959]) (Hint: use the fact that every r.e. set has infinitely many indices.)
b) *The m-degree of an index set is irreducible.* (Rogers [1959])

Complexity of index sets

Index sets of low complexity can be classified according to their position in the Arithmetical Hierarchy.

- From the Rice Theorem II.2.9, it follows that $\theta\mathcal{A}$ *is recursive if and only if it is trivial*, i.e. either empty or containing all r.e. sets.
- From the Myhill-Shepherdson Theorem II.4.2, it follows that $\theta\mathcal{A}$ *is r.e. if and only if it is effectively open* in the positive information topology of Section II.3.
- Selivanov [1982a] has defined, for families of r.e. sets, an analogue $\widehat{\Sigma}_n^0$ of the Arithmetical Hierarchy (where $\widehat{\Sigma}_1^0$ is the class of effectively open families), and has proved that $\theta\mathcal{A} \in \Sigma_n^0$ *if and only if* $\mathcal{A} \in \widehat{\Sigma}_n^0$, *for every* $n \geq 3$. He has also shown that in a sense no criterion exists for $n = 2$. Extensions to the hyperarithmetical levels are in Selivanov [1984a].

A more general measure of complexity for index sets is their Turing degree.

Proposition X.9.3 (Rice [1953], Hay [1973]) *A degree $\boldsymbol{a}$ contains a nontrivial index set if and only if $\boldsymbol{a} \geq \mathbf{0'}$.*

Proof. Let $\mathcal{A}$ be a nontrivial class of r.e. sets. Suppose $\emptyset \notin \mathcal{A}$ (otherwise consider $\overline{\mathcal{A}}$), and let $A \in \mathcal{A}$. If f is a recursive function such that

$$\mathcal{W}_{f(e)} = \begin{cases} A & \text{if } e \in \mathcal{K} \\ \emptyset & \text{otherwise,} \end{cases}$$

then

$$e \in \mathcal{K} \Leftrightarrow \mathcal{W}_{f(e)} \in \mathcal{A} \Leftrightarrow f(e) \in \theta\mathcal{A}.$$

It follows that $\boldsymbol{a}$ is the degree of $\theta\mathcal{A}$, then $\boldsymbol{a} \geq \mathbf{0'}$.

Conversely, let $\boldsymbol{a}$ be a degree such that $\boldsymbol{a} \geq \mathbf{0'}$. By the Jump Inversion Theorem V.2.24, there is a set A such that $A' \in \boldsymbol{a}$. Consider

$$x \in B \Leftrightarrow \mathcal{W}_x \cap A' \neq \emptyset.$$

Then $A' \leq_m B$ via a recursive f such that $\mathcal{W}_{f(x)} = \{x\}$, and $B \leq_m A'$ because B is r.e. in A. Thus $A' \equiv_m B$. By X.9.2.b, A' is an index set. $\square$

The proof shows in particular that every jump is an index set (up to recursive isomorphism).

Exercise X.9.4 *For $\boldsymbol{a} \geq \mathbf{0}'$, every countable partial ordering is embeddable in the m-degrees of index sets contained in $\boldsymbol{a}$.* (Hay [1973]) (Hint: for $\mathbf{0}'$ use the following facts: every countable partial ordering is embeddable in the low degrees; $A \leq_T B$ if and only if $A' \leq_1 B'$; jumps are index sets. In general, use relativization and the Jump Inversion Theorem.)

Hay [1972], [1973a] has more results about the m-degrees of index sets inside Turing degrees.

Specific index sets

We now study a number of natural index sets of increasing complexity. Interestingly, they all turn out to be complete at the appropriate level of the (relativized) Arithmetical Hierarchy (although, by X.9.3, this is not true for arbitrary index sets).

Proposition X.9.5 (Dekker and Myhill [1958a]) *The set*

$$\mathbf{Em} = \{x : \mathcal{W}_x = \emptyset\}$$

is Π_1^0-complete. Similarly,

$$\mathbf{Nem} = \{x : \mathcal{W}_x \neq \emptyset\}$$

is Σ_1^0-complete.

Proof. $\{x : \mathcal{W}_x \neq \emptyset\}$ is Σ_1^0 because

$$\mathcal{W}_x \neq \emptyset \;\Leftrightarrow\; (\exists z)(z \in \mathcal{W}_x).$$

It is Σ_1^0-complete because, given $A \in \Sigma_1^0$, if f is a recursive function such that

$$\mathcal{W}_{f(x)} = \begin{cases} \omega & \text{if } x \in A \\ \emptyset & \text{otherwise,} \end{cases}$$

then

$$x \in A \;\Leftrightarrow\; \mathcal{W}_{f(x)} \neq \emptyset.$$

The second assertion follows from the first, by taking complements. $\square$

A number of incompleteness results for formal systems can be derived from results on index sets, see Hájek [1979]. For example, from X.9.5 it follows that *if $\mathcal{F}$ is any formal system proving only true facts about r.e. sets, then there is x such that $\mathcal{W}_x = \emptyset$ is true, but cannot be proved in $\mathcal{F}$.* Indeed,

$$\{x : \text{`}\mathcal{W}_x = \emptyset\text{' is provable in } \mathcal{F}\}$$

is Σ_1^0 because $\mathcal{F}$ is a formal system, while $\{x : \mathcal{W}_x = \emptyset\}$ is Π_1^0-complete, and hence not Σ_1^0.

Proposition X.9.6 (Turing [1939], Dekker and Myhill [1958a]) *The sets*

$$\mathbf{Fin} = \{x : \mathcal{W}_x \textit{ finite}\} \qquad \textit{and} \qquad \mathbf{Ntot} = \{x : \mathcal{W}_x \neq \omega\}$$

are Σ_2^0-complete. Similarly,

$$\mathbf{Inf} = \{x : \mathcal{W}_x \textit{ infinite}\} \qquad \textit{and} \qquad \mathbf{Tot} = \{x : \mathcal{W}_x = \omega\}$$

are Π_2^0-complete.

Proof. **Inf** is Π_2^0 because

$$\mathcal{W}_x \text{ infinite } \Leftrightarrow (\forall y)(\exists z)(z > y \wedge z \in \mathcal{W}_x).$$

Similarly, **Tot** is Π_2^0.

Let $A \in \Pi_2^0$ be given. Then there is R recursive such that

$$x \in A \Leftrightarrow (\forall y)(\exists z)R(x, y, z).$$

By letting f be a recursive function such that

$$y \in \mathcal{W}_{f(x)} \Leftrightarrow (\exists z)R(x, y, z)$$

one has

$$x \in A \Leftrightarrow \mathcal{W}_{f(x)} = \omega,$$

and thus **Tot** is Π_2^0-complete. A little modification is needed to show that so is **Inf**. Let g be recursive function such that

$$s \in \mathcal{W}_{g(x)} \Leftrightarrow (\forall y \leq s)(\exists z)R(x, y, z).$$

Then $\mathcal{W}_{g(x)}$ is ω if $x \in A$, and is finite otherwise. In particular,

$$x \in A \Leftrightarrow \mathcal{W}_{g(x)} \text{ infinite.}$$

The second assertions follow from the first, by taking complements. $\Box$

Exercises X.9.7 Finite families of r.e. sets. The index sets of finite families are completely described by the following exercises.

a) *If B is finite, then $\theta\{B\}$ has degree $\mathbf{0}'$, and has two possible m-degrees, depending on whether B is empty or not. If B is infinite, then $\theta\{B\}$ is Π^0_2-complete.* (Hay [1966]) (Hint: if B is infinite, modify X.9.6 by letting

$$s \in \mathcal{W}_{g(x)} \Leftrightarrow s \in B \wedge (\forall y \leq s)(\exists z)R(x,y,z).$$

If $B = \emptyset$, then its index set has the m-degree of $\overline{\mathcal{K}}$ by X.9.5. If $B \neq \emptyset$ is finite, let

$$\mathcal{W}_{f(x)} = \begin{cases} \omega & \text{if } (x)_2 \in \mathcal{K} \\ B & \text{if } (x)_1 \in \mathcal{K} \wedge (x)_2 \notin \mathcal{K} \\ \emptyset & \text{if } (x)_1 \notin \mathcal{K}. \end{cases}$$

Then $x \in \mathcal{K} \cdot \overline{\mathcal{K}}$ if and only if $\mathcal{W}_{f(x)} = B$. Conversely, the set $\{x : \mathcal{W}_x = B\}$ is easily reduced to $\mathcal{K} \cdot \overline{\mathcal{K}}$, since $B \subseteq \mathcal{W}_x$ is Σ^0_1 (as B is finite), and $\mathcal{W}_x \subseteq B$ is Π^0_1.)

b) *If $\mathcal{A}$ is a finite family of finite sets, then $\theta\mathcal{A}$ has degree $\mathbf{0}'$. The m-degrees of such index sets are linearly ordered with order type ω.* (Ershov [1968], Hay [1969]). (Hint: extend the proof of part a), by considering the longest possible chain of elements of $\mathcal{A}$ ordered by the following relation:

$$A \sqsubseteq B \Leftrightarrow (\exists D \text{ finite})(D \notin \mathcal{A} \wedge A \subseteq D \subseteq B).$$

Consider also the two cases $\emptyset \in \mathcal{A}$ and $\emptyset \notin \mathcal{A}$. The possible m-degrees are those of

$$\overline{\mathcal{K}} \quad \mathcal{K} \cdot \overline{\mathcal{K}} \quad (\overline{\mathcal{K}} + \mathcal{K}) \cdot \overline{\mathcal{K}} \quad \mathcal{K} \cdot \overline{\mathcal{K}} + \mathcal{K} \cdot \overline{\mathcal{K}} \quad \cdots$$

where

$$\langle x,y \rangle \in A \cdot B \Leftrightarrow x \in A \wedge y \in B \qquad \text{and} \qquad \langle x,y \rangle \in A + B \Leftrightarrow x \in A \vee y \in B.$$

In other words, we obtain a zig-zag picture of the complete sets of the finite levels of the Boolean Hierarchy, see IV.1.18.c.)

c) *If $\mathcal{A}$ is a finite family of r.e. sets containing at least one infinite member, then $\theta\mathcal{A}$ is Π^0_2-complete.* (Shapiro [1956]) (Hint: let $B \in \mathcal{A}$ be infinite. If there are no finite sets in $\mathcal{A}$, the proof of part a) suffices. If there are finite sets, let $C \subseteq B$ be a finite set with more elements than any of the finite members of $\mathcal{A}$, and let

$$s \in \mathcal{W}_{g(x)} \Leftrightarrow s \in C \vee [s \in B \wedge (\forall y \leq s)(\exists z)R(x,y,z)].$$

For more on this topic and extensions of results, see Hay [1972], [1974], [1974a], [1975], [1976], Grassin [1974], Hay, Manaster and Rosenstein [1975], Johnson [1978].

Exercises X.9.8 Families of finite sets.

a) *If $\mathcal{A}$ is a strong array, then $\theta\mathcal{A}$ is either of degree $\mathbf{0}'$ or Σ^0_2-complete. The m-degrees of such index sets form a complicated structure, which isomorphically embeds the m-degrees of simple sets.* (Selivanov [1978], [1978a], Johnson [1978], Brandt [1988]) (Hint: if $\mathcal{A} = \{D_{f(x)}\}_{x \in \omega}$, then $\{x : (\exists z)(\mathcal{W}_x = D_{f(z)})\}$ is Σ^0_2. If there is an infinite r.e. chain $A_0 \subset A_1 \subset \cdots$ of elements of $\mathcal{A}$, then $\theta\mathcal{A}$ is Σ^0_2-complete, because **Fin** is reduced to it by any recursive function h such that

$$|\mathcal{W}_{x,s}| = n \ \Rightarrow\ \mathcal{W}_{h(x),s} = A_n.$$

If there is no such r.e. chain, then $\theta\mathcal{A} \in \Pi^0_2$ (and hence $\theta\mathcal{A} \in \Delta^0_2$, and it has degree $\mathbf{0}'$ by X.9.3):

$$\mathcal{W}_x \in \mathcal{A} \ \Leftrightarrow\ (\forall n)[D_n \subseteq \mathcal{W}_x \ \Rightarrow\ (\exists z)(D_n \subseteq D_{f(z)} \subseteq \mathcal{W}_x)].$$

Given A, let $\mathcal{A}_A = \{\emptyset, \{a\}\}_{a\in A}$. If A is r.e., then $\mathcal{A}_A$ is a strong array. If f reduces A to B, then the function g defined as follows reduces $\theta\mathcal{A}_A$ to $\theta\mathcal{A}_B$:

$$\mathcal{W}_{g(x)} = \begin{cases} \emptyset & \text{if } \mathcal{W}_x = \emptyset \\ \{f(a)\} & \text{if } \mathcal{W}_x = \{a\} \\ \omega & \text{otherwise.} \end{cases}$$

Conversely, if g reduces $\theta\mathcal{A}_A$ to $\theta\mathcal{A}_B$, let $\mathcal{W}_{h(x)} = \{x\}$. Then gh reduces A to $\theta\mathcal{A}_B$. If A is simple, then

$$\{x : \mathcal{W}_{gh(x)} \text{ is nonempty but not a singleton}\}$$

is finite. So, except for finitely many x, either $x \in A$ or $\mathcal{W}_{gh(x)}$ is a singleton. Then the following f reduces A to B:

$$f(x) = \begin{cases} b & \text{if } x \text{ shows up first in } A \\ y & \text{if } y \text{ shows up first in } \mathcal{W}_{gh(x)}, \end{cases}$$

where $b \in B$.)

b) *A Turing degree contains the index set of a weak array if and only if it is r.e. in and above* $\mathbf{0}'$. (Selivanov [1978]) (Hint: if f enumerates a weak array, then $\{x : (\exists z)(\mathcal{W}_x = \mathcal{W}_{f(z)})\}$ is Σ^0_2, and hence of degree r.e. in $\mathbf{0}'$. By X.9.3, its degree must also be above $\mathbf{0}'$. Now let $\mathcal{K} \leq_T A$ and $A \in \Sigma^0_2$. Since **Fin** is Σ^0_2-complete, A is reduced to it by some recursive g. Let $\mathcal{A}$ be the family of the following r.e. sets:

$$B^s_x = \begin{cases} \{0, x+1\} & \text{if } |\mathcal{W}_{g(x)}| > s \\ \{x+1\} & \text{otherwise.} \end{cases}$$

Then $A \equiv_T \theta\mathcal{A}$. One direction follows from

$$x \in A \ \Leftrightarrow\ \mathcal{W}_{g(x)} \text{ finite} \ \Leftrightarrow\ \{x+1\} \in \mathcal{A}.$$

The other comes from the fact that $\mathcal{K} \leq_T A$, since then given $\mathcal{W}_x$ we can, recursively in $\mathcal{K}$ and hence in A, first see whether $\mathcal{W}_x$ has 1 or 2 elements, and then determine which ones.)

c) *A Turing degree contains the index set of a family of finite sets if and only if it is above* $\mathbf{0}'$. (Hint: consider $\mathcal{A}_A = \{\{a\} : a \in A\}$. If $\mathcal{K} \leq_T A$, then $A \equiv_T \theta\mathcal{A}_A$ as in part b).)

Exercises X.9.9 (Hay [1965]) a) *If A and B are distinct r.e. sets, their index sets are effectively inseparable.* (Hint: given x and y, let

$$\mathcal{W}_{f(x,y,z)} = \begin{cases} A & \text{if } z \text{ shows up first in } \mathcal{W}_y \\ B & \text{if } z \text{ shows up first in } \mathcal{W}_x \\ \emptyset & \text{if } z \notin \mathcal{W}_x \cup \mathcal{W}_y. \end{cases}$$

By the Fixed-Point Theorem with parameters (II.2.11) there is a recursive function g such that

$$\mathcal{W}_{g(x,y)} = \mathcal{W}_{f(x,y,g(x,y))}.$$

Then $g(x,y) \notin \mathcal{W}_x \cup \mathcal{W}_y$ if $A \subseteq \mathcal{W}_x$, $B \subseteq \mathcal{W}_y$ and $\mathcal{W}_x \cap \mathcal{W}_y = \emptyset$.)

b) *Two disjoint r.e. sets are effectively inseparable if and only if each of them contains a set recursively isomorphic to an index set.* (Hint: one direction comes from part a). Since all pairs of effectively inseparable r.e. sets are recursively isomorphic by III.7.15, it is enough to exhibit one pair containing index sets. Let $x \neq y$ and define

$$z \in A \ \Leftrightarrow\ x \in \mathcal{W}_z \qquad \text{and} \qquad z \in B \ \Leftrightarrow\ y \in \mathcal{W}_z.$$

Let A_0 and B_0 be disjoint subsets of A and B, respectively, with union $A \cup B$, by II.1.23. Then, for example, A_0 contains $\theta\{x\}$ and B_0 contains $\theta\{y\}$, and they are effectively inseparable by part a).)

Hay [1966], [1969] provides a classification of the recursive isomorphism types of pairs of index sets. Ershov [1970b] does the same for n-tuples of index sets.

The next result provides a general method of generating index sets of increasing complexity.

Proposition X.9.10 (McLaughlin [1971], Hartmanis and Lewis [1971], Jockusch) *For $n \geq 1$, if A is Π^0_n-complete then its* **weak jump**, *defined (in IX.4.4) as*

$$H_A = \{x : \mathcal{W}_x \cap A \neq \emptyset\},$$

is Σ^0_{n+1}-complete.

Proof. If $A \in \Pi^0_n$ then $\{x : \mathcal{W}_x \cap A \neq \emptyset\} \in \Sigma^0_{n+1}$. Now let $B \in \Sigma^0_{n+1}$ and let $R \in \Pi^0_n$ be such that

$$x \in B \ \Leftrightarrow\ (\exists t)R(x,t).$$

Since A is Π^0_n-complete, for some recursive g

$$R(x,t) \ \Leftrightarrow\ g(x,t) \in A.$$

Let f be a recursive function such that

$$u \in \mathcal{W}_{f(x)} \ \Leftrightarrow\ (\exists t)(g(x,t) = u).$$

Then

$$\begin{array}{rcl} x \in B & \Leftrightarrow & (\exists t) R(x,t) \\ & \Leftrightarrow & (\exists t)(g(x,t) \in A) \\ & \Leftrightarrow & (\exists u)(u \in \mathcal{W}_{f(x)} \ \wedge \ u \in A) \\ & \Leftrightarrow & \mathcal{W}_{f(x)} \cap A \neq \emptyset. \quad \Box \end{array}$$

In particular, $\{x : \mathcal{W}_x \cap \mathbf{Inf} \neq \emptyset\}$ is Σ^0_3-complete and $\{x : \mathcal{W}_x \subseteq \mathbf{Fin}\}$ is Π^0_3-complete (by X.9.6).

Proposition X.9.11 (Rogers [1959]) *The set*

$$\mathbf{Cof} = \{x : \mathcal{W}_x \textit{ cofinite}\}$$

is Σ^0_3-complete.

Proof. **Cof** is Σ^0_3, because

$$\mathcal{W}_x \text{ cofinite } \Leftrightarrow (\exists y)(\forall z \geq y)(z \in \mathcal{W}_x).$$

Since $\{x : \mathcal{W}_x \cap \mathbf{Inf} \neq \emptyset\}$ is Σ^0_3-complete, it is enough to obtain f recursive such that

$$\begin{array}{rcl} \mathcal{W}_x \cap \mathbf{Inf} \neq \emptyset & \Rightarrow & \mathcal{W}_{f(x)} \text{ cofinite} \\ \mathcal{W}_x \cap \mathbf{Inf} = \emptyset & \Rightarrow & \mathcal{W}_{f(x)} \text{ nonrecursive.} \end{array}$$

We fix x and build an r.e. set A_x, uniformly in x. The existence of f will then follow by the S^m_n-Theorem. Since we do not know whether $\mathcal{W}_x \cap \mathbf{Inf} \neq \emptyset$ or not, we will have to work in both directions.

The first type of requirement is

$$R_{2e} \ : \ A_x \neq \overline{\mathcal{W}}_e.$$

To satisfy R_{2e} we choose a witness z_e and put it into A_x if and only if it shows up in $\mathcal{W}_e$. If every R_{2e} is satisfied, A_x is not recursive.

The second type of requirement is

$$R_{2e+1} \ : \ e \in \mathcal{W}_x \ \Rightarrow \ (\forall z)(\forall s)(z < \max \mathcal{W}_{e,s} \ \Rightarrow \ z \in A_{x,s}).$$

If $e \in \mathcal{W}_x \cap \mathbf{Inf}$ and R_{2e+1} is satisfied, then every element goes into A_x.

The construction is as follows. We start with $A_{x,0} = \emptyset$. At stage $s+1$, $\overline{A}_{x,s} = \{z^s_0 < z^s_1 < \cdots\}$. By construction $A_{x,s}$ is finite, so when we move witnesses we can still send them to elements of $\overline{A}_{x,s}$. For each $e \leq s$ such that $z^s_e \notin A_{x,s}$, we put it into A_x if and only if it is in $\mathcal{W}_{e,s}$. To satisfy R_{2e+1}

we would like to put into A_x every element less than max $\mathcal{W}_{e,s}$, if $e \in \mathcal{W}_{x,s}$. For this it would be enough to move all markers z_i^s to positions larger than max $\mathcal{W}_{e,s}$ (since everything which is not a marker goes into A_x). But since there are conditions of higher priority, we just move above max $\mathcal{W}_{e,s}$ all markers z_i^s for $i > e$, if $e \in \mathcal{W}_{x,s}$.

If $\mathcal{W}_x \cap \mathbf{Inf} \neq \emptyset$, $\mathcal{W}_e$ is infinite for some $e \in \mathcal{W}_x$. Then every marker z_i^s with $i > e$ moves infinitely many times and A_x is cofinite.

If $\mathcal{W}_x \cap \mathbf{Inf} = \emptyset$, $\mathcal{W}_e$ is finite for every $e \in \mathcal{W}_x$. Let s_e be a stage such that $\mathcal{W}_{e,s_e} = \mathcal{W}_e$. Then

$$(\forall s \geq s_e)(\forall i > e)(z_i^s \text{ does not move for the sake of } \mathcal{W}_e).$$

Since z_i^s only moves for the sake of $\mathcal{W}_e$ when $e < i$, each marker settles down and the construction produces a nonrecursive A_x. $\square$

The proof actually shows that *every Σ_3^0 class of recursive sets containing all the cofinite sets is Σ_3^0-complete*, since $\mathcal{W}_{f(x)}$ is either cofinite or nonrecursive by construction. The following is a special interesting case.

Corollary X.9.12 (Rogers [1959], Mostowski, Shapiro, Ehrenfeucht) *The set*

$$\mathbf{Rec} = \{x : \mathcal{W}_x \textit{ recursive}\}$$

is Σ_3^0-complete.

Proof. Rec is Σ_3^0 because

$$\begin{aligned} \mathcal{W}_x \text{ recursive } &\Leftrightarrow (\exists e)(\mathcal{W}_x = \overline{\mathcal{W}}_e) \\ &\Leftrightarrow (\exists e)(\forall z)(z \in \mathcal{W}_x \Leftrightarrow z \notin \mathcal{W}_e). \quad \square \end{aligned}$$

It follows from this and X.9.10 that $\{x : \mathcal{W}_x \subseteq \mathbf{Rec}\}$ is Π_4^0-complete.

Exercises X.9.13 a) *For $n \geq 1$, if A is Π_n^0-complete, then $\{x : \mathcal{W}_x \cap A$ finite$\}$ is Σ_{n+2}^0-complete.* (Kreisel, Shoenfield and Wang [1960], Hartmanis and Lewis [1971]) (Hint: given $B \in \Pi_{n+2}^0 = \Pi_3^{0,C}$ with $C \in \Sigma_{n-1}^0$, by relativizing the proof of X.9.11 we obtain a recursive function f such that

$$x \in B \Leftrightarrow \mathcal{W}_{f(x)}^C \text{ coinfinite} \Leftrightarrow \{z : z \notin \mathcal{W}_{f(x)}^C\} \text{ infinite.}$$

By Π_n^0-completeness of A, there is a recursive one-one function h such that

$$z \notin \mathcal{W}_{f(x)}^C \Leftrightarrow h(x,z) \in A.$$

Then, if $\mathcal{W}_{g(x)} = \{u : (\exists z)(h(x,z) = u)\}$, we have

$$\{z : z \notin \mathcal{W}_{f(x)}^C\} \text{ infinite} \Leftrightarrow \mathcal{W}_{g(x)} \cap A \text{ infinite,}$$

by one-oneness of h.)

b) *The set $\{x : \mathcal{W}_x \cap \mathbf{Inf}$ finite$\}$ is Σ_4^0-complete.*

Exercises X.9.14 In the following exercises $\leq_r$ stands for any of the standard reducibilities caught in between $\leq_m$ and $\leq_{wtt}$.

a) *The set* $\{x : \mathcal{W}_x$ *creative*$\}$ *is* Σ^0_3*-complete.* (Rogers [1959]) (Hint: by the methods of X.9.11 one can build, given x, r.e. sets A_x and B_x such that

$$\begin{array}{lcll} \mathcal{W}_x \cap \mathbf{Inf} \neq \emptyset & \Rightarrow & A_x =^* \mathcal{K} & \text{(so } A_x \text{ is creative)} \\ \mathcal{W}_x \cap \mathbf{Inf} = \emptyset & \Rightarrow & B_x \not\leq_T A_x & \text{(so } A_x \text{ is not } T\text{-complete).)} \end{array}$$

b) *If* C *is an r.e. set and* $C \neq \emptyset, \omega$*, then* $\{x : \mathcal{W}_x \equiv_r C\}$ *is* Σ^0_3*-complete.* (Yates [1966a]) (Hint: by the methods of X.9.11 one can build, given x, an r.e. set A_x such that

$$\begin{array}{lcll} \mathcal{W}_x \cap \mathbf{Inf} \neq \emptyset & \Rightarrow & A_x =^* C & \text{(so } A_x \equiv_m C) \\ \mathcal{W}_x \cap \mathbf{Inf} = \emptyset & \Rightarrow & C \not\leq_T A_x. & \end{array}$$

The requirements for the second conditions are handled by the Sacks agreement method. This works if C is nonrecursive. The case of C recursive follows from X.9.12.)

c) *If* C *is an r.e. nonrecursive set, then* $\{x : C \leq_r \mathcal{W}_x\}$ *is* Σ^0_3*-complete.* (Kallibekov [1971]) (Hint: see part b).)

d) *If* C *is an r.e.* r*-incomplete set, then* $\{x : \mathcal{W}_x \leq_r C\}$ *is* Σ^0_3*-complete.* (Kallibekov [1971]) (Hint: by the methods of X.9.11 one can build, given x, an r.e. set A_x such that

$$\begin{array}{lcl} \mathcal{W}_x \cap \mathbf{Inf} \neq \emptyset & \Rightarrow & A_x =^* C \\ \mathcal{W}_x \cap \mathbf{Inf} = \emptyset & \Rightarrow & A_x \not\leq_r C.) \end{array}$$

e) *If* C *is an r.e. nonrecursive* r*-incomplete set, then* $\{x : \mathcal{W}_x|_r C\}$ *is* Π^0_3*-complete.* (Kallibekov [1971])

f) *The class of r.e. sets* r*-reducible to a given r.e. set is recursively enumerable.* (Kallibekov [1971]) (Hint: by II.5.25.)

Herrmann [1986] has proved that $\{x : \mathcal{W}_x \equiv_1 C\}$ *is* Σ^0_3*-complete*, for any infinite coinfinite r.e. set C.

The proof of X.9.12 (obtained as a corollary to X.9.11) used the priority method. A direct, priority-free proof (useful for extensions in X.9.15) is as follows. If $B \in \Sigma^0_3$, there exists a recursive R such that

$$x \in B \ \Leftrightarrow \ (\exists y)(\forall t)(\exists s)R(x, y, t, s).$$

We want a recursive function h such that

$$\begin{array}{lcll} x \in B & \Rightarrow & \mathcal{W}_{h(x)} \text{ recursive} & \\ x \in \overline{B} & \Rightarrow & \mathcal{K} \leq_T \mathcal{W}_{h(x)} & \quad \text{(in particular, } \mathcal{W}_{h(x)} \text{ nonrecursive).} \end{array}$$

The idea is to unfold $\mathcal{W}_{h(x)}$ and let its e-th column be equal to ω exactly when either $e \in \mathcal{K}$ or, for some $y \leq e$, $(\forall t)(\exists s)R(x, y, t, s)$, and finite otherwise. Thus let

$$\langle e, u \rangle \in \mathcal{W}_{h(x)} \ \Leftrightarrow e \in \mathcal{K} \ \vee \ (\exists y \leq e)(\forall t \leq u)(\exists s)R(x, y, t, s).$$

- If $x \in B$, then $(\forall t)(\exists s)R(x,y,t,s)$ for some y. So
$$(\forall e \geq y)(\mathcal{W}_{h(x)}^{[e]} = \omega),$$
and $\mathcal{W}_{h(x)}$ is recursive.

- If $x \in \overline{B}$, then no $\mathcal{W}_{h(x)}^{[e]}$ is equal to ω due to the second clause of the definition. So
$$e \in \overline{\mathcal{K}} \iff \mathcal{W}_{h(x)}^{[e]} \neq \omega.$$
But $\mathcal{K}$ is r.e. and $\overline{\mathcal{K}}$ is r.e. in $\mathcal{W}_{h(x)}$, since
$$e \in \overline{\mathcal{K}} \iff \mathcal{W}_{h(x)}^{[e]} \neq \omega \iff (\exists u)(\langle e,u\rangle \notin \mathcal{W}_{h(x)}).$$
Thus $\mathcal{K} \leq_T \mathcal{W}_{h(x)}$.

The proof just given is easily adapted to show that if
$$x \in B \iff (\exists y)(\forall t)Q(x,y,t)$$
with Q r.e. and recursive in an r.e. set C, then there is a recursive function h such that
$$\begin{aligned} x \in B &\quad\Rightarrow\quad \mathcal{W}_{h(x)} \text{ recursive} \\ x \in \overline{B} &\quad\Rightarrow\quad C \equiv_T \mathcal{W}_{h(x)}. \end{aligned}$$
Indeed, it is enough to let
$$\langle e,u\rangle \in \mathcal{W}_{h(x)} \iff e \in C \lor (\exists y \leq e)(\forall t \leq u)Q(x,y,t).$$
The hypothesis that Q is r.e. is required to have $\mathcal{W}_{h(x)}$ r.e. The hypothesis that Q is recursive in C gives instead $\mathcal{W}_{h(x)} \leq_T C$.

Note that if C is r.e., then every $\Sigma_2^{0,C}$ set B can be represented as above. Indeed, a typical $\Pi_1^{0,C}$ set has, by II.3.11, the form $(\forall z)R(\ldots,\hat{c}_C(z))$, which is equivalent to
$$(\forall z)(\forall s_0)(\exists s > s_0)R(\ldots,\hat{c}_{C_s}(z)).$$
Now, the part $(\exists s > s_0)R(\ldots,\hat{c}_{C_s}(z))$ is r.e. by definition, and is recursive in C because it is equivalent to $R(\ldots,\hat{c}_{C_s}(z))$ for any s such that $C_s[z] = C[z]$.

Proposition X.9.15 (Yates [1966a]) *For any r.e. set C, the set*
$$\{x : \mathcal{W}_x \equiv_T C\}$$
is $\Sigma_3^{0,C}$-complete.

Proof. Such a set is $\Sigma_3^{0,C}$ because

$$\begin{aligned} \mathcal{W}_x \leq_T C &\Leftrightarrow (\exists e)(\overline{\mathcal{W}}_x = \mathcal{W}_e^C) \\ &\Leftrightarrow (\exists e)(\forall z)(z \notin \mathcal{W}_x \Leftrightarrow z \in \mathcal{W}_e^C), \end{aligned}$$

and similarly for $C \leq_T \mathcal{W}_x$.

Given $A \in \Sigma_3^{0,C}$, we want a recursive function f such that

$$x \in A \Leftrightarrow \mathcal{W}_{f(x)} \equiv_T C.$$

Since $A \in \Sigma_3^{0,C}$, there is $B \in \Sigma_2^{0,C}$ such that

$$x \in \overline{A} \Leftrightarrow (\forall e)B(x,e).$$

By applying the remarks given before the statement of the result we can obtain, uniformly in e, an r.e. set $\mathcal{W}_{g(x)} = \bigoplus_{e\in\omega} \mathcal{W}_{h(x,e)}$ with the following properties:

$$\begin{aligned} B(x,e) &\Rightarrow \mathcal{W}_{g(x)}^{[e]} \text{ recursive} \\ \neg B(x,e) &\Rightarrow C \equiv_T \mathcal{W}_{g(x)}^{[e]}. \end{aligned}$$

Then

$$\begin{aligned} x \in A &\Rightarrow (\exists e)[(\forall i < e)(\mathcal{W}_{g(x)}^{[i]} \text{ recursive}) \wedge \mathcal{W}_{g(x)}^{[e]} \equiv_T C] \\ x \in \overline{A} &\Rightarrow (\forall e)[\mathcal{W}_{g(x)}^{[e]} \text{ recursive}]. \end{aligned}$$

We can suppose that C is not recursive, since this case has already been proved in X.9.12. We would simply like to take $f = g$, but if e.g. $x \in \overline{A}$ then, while $\mathcal{W}_{g(x)}$ is piecewise recursive, it is not necessarily $\mathcal{W}_{g(x)} \not\equiv_T C$. But by applying the strong Thickness Lemma (X.3.9) uniformly in x we can obtain a thick subset $\mathcal{W}_{f(x)}$ of $\mathcal{W}_{g(x)}$.

- If $x \in \overline{A}$, then $\mathcal{W}_{g(x)}$ is piecewise recursive. So $(\forall e)(C \not\leq_T \mathcal{W}_{g(x)}^{[e]})$ and thus $C \not\leq_T \mathcal{W}_{f(x)}$.
- If $x \in A$, let e be such that

  $$(\forall i < e)(\mathcal{W}_{g(x)}^{[i]} \text{ recursive}) \wedge \mathcal{W}_{g(x)}^{[e]} \equiv_T C.$$

  Then $C \not\leq_T \mathcal{W}_{g(x)}^{[<e]}$. By X.3.8, $\mathcal{W}_{f(x)}^{[e]} =^* \mathcal{W}_{g(x)}^{[e]}$. So

  $$\begin{gathered} C \leq_T \mathcal{W}_{g(x)}^{[e]} \leq_T \mathcal{W}_{f(x)}^{[e]} \leq_T \mathcal{W}_{f(x)} \\ \mathcal{W}_{f(x)} \leq_T \mathcal{W}_{g(x)} \leq_T C, \end{gathered}$$

  where the last inequality holds by the strong Thickness Lemma X.3.9. Thus $\mathcal{W}_{f(x)} \equiv_T C$ as required. □

Corollary X.9.16 *The set $\{x : \mathcal{W}_x$ T-complete$\}$ is Σ_4^0-complete.*

The proof given above is a direct consequence of the Thickness Lemma, and it is due to Shoenfield [1971a].

The only case in which the (infinite injury) priority method is required to prove the result is when $C \equiv_T \mathcal{K}$, i.e. for the corollary. In the other cases, i.e. if C is an r.e. T-incomplete set, then X.9.17.c gives a priority-free proof of the fact that $\{x : \mathcal{W}_x \leq_T C\}$ is $\Sigma_3^{0,C}$-complete. From this the theorem follows, since if $\mathcal{W}_{g(x)} = \mathcal{W}_x \oplus C$, then $\mathcal{W}_x \leq_T C$ if and only if $\mathcal{W}_{g(x)} \equiv_T C$.

Exercises X.9.17 a) *If C is an r.e. nonrecursive set, then $\{x : C \leq_T \mathcal{W}_x\}$ is Σ_4^0-complete.* (Yates [1969]) (Hint: for $C \equiv_T \mathcal{K}$ this follows from the corollary above. For $C <_T \mathcal{K}$, the proof of X.9.15 can be modified to obtain, given $A \in \Sigma_4^0 = \Sigma_3^{0,\mathcal{K}}$, a recursive function f such that

$$\begin{array}{lll} x \in A & \Rightarrow \quad \mathcal{W}_{f(x)} \equiv_T \mathcal{K} & \Rightarrow \quad C \leq_T \mathcal{W}_{f(x)} \\ x \in \overline{A} & \Rightarrow \quad C \not\leq_T \mathcal{W}_{f(x)}.) & \end{array}$$

b) *If C is an r.e. nonrecursive and T-incomplete set, then $\{x : \mathcal{W}_x |_T C\}$ is Π_4^0-complete.* (Yates [1969]) (Hint: let B be an r.e. set incomparable with C. Given $A \in \Pi_4^0$, i.e. $\overline{A} \in \Sigma_4^0$, modify part a) to obtain a recursive function f such that $B \leq_T \mathcal{W}_{f(x)}$ and

$$\begin{array}{lll} x \in \overline{A} & \Rightarrow \quad \mathcal{W}_{f(x)} \equiv_T \mathcal{K} & \Rightarrow \quad C \leq_T \mathcal{W}_{f(x)} \\ x \in A & \Rightarrow \quad C \not\leq_T \mathcal{W}_{f(x)}. & \end{array}$$

Then $\mathcal{W}_{f(x)}$ is Turing incomparable with C if $x \in A$, otherwise $B \leq_T \mathcal{W}_{f(x)} \leq_T C$.)

c) *If C is an r.e. T-incomplete set, then $\{x : \mathcal{W}_x \leq_T C\}$ is $\Sigma_3^{0,C}$-complete.* (Yates [1969]). (Hint: given $A \in \Sigma_3^{0,C}$, find a recursive function f such that

$$x \in A \Leftrightarrow \mathcal{W}_{f(x)} \leq_T C.$$

Use the coding method to attempt a coding of $\mathcal{K}$ into $\mathcal{W}_{f(x)}^{[e]}$ below the length of agreement of $\mathcal{W}_{f(x)}$ and $\{e\}^C$, whenever it looks that $(\exists y)(\forall z)Q(x, y, z, e)$, where $x \in \overline{A} \Leftrightarrow (\forall e)(\exists y)(\forall z)Q(x, y, z, e)$. Thus if $x \in \overline{A}$, such a y would really exist, and if $\mathcal{W}_{f(x)} \simeq \{e\}^C$, the coding would succeed, contradicting $\mathcal{K} \not\leq_T C$.)

d) *The class of r.e. sets recursive in C is recursively enumerable if and only if $C \equiv_T \mathcal{K}$ or C is low$_2$.* (Yates [1969]) (Hint: such a class contains all finite sets, and by II.5.25 it is recursively enumerable if and only if it is Σ_3^0, i.e. if and only if $\Sigma_3^{0,C} = \Sigma_3^0$.)

Exercises X.9.18 Here we classify index sets relative to the jump classes introduced in XI.1.9. For convenience of notation, we identify a set and its degree.

a) $\{x : \mathcal{W}_x \in \mathbf{L}_n\}$ *is Σ_{n+3}^0-complete.* (Schwarz [199?]) (Hint: by X.9.12 we have $\emptyset^{(3)} \leq_m \{x : \mathcal{W}_x \in \mathbf{L}_0\}$. By relativization, we obtain $f \leq_T \emptyset^{(n)}$ such that

$$x \in \emptyset^{(n+3)} \Leftrightarrow \mathcal{W}_{f(x)}^{\emptyset^{(n)}} \equiv_T \emptyset^{(n)}$$

where, for all x, $\emptyset^{(n)} \leq_T \mathcal{W}_{f(x)}^{\emptyset^{(n)}}$. By the Sacks Jump Inversion Theorem relativized (XI.1.23) there is a recursive g such that if $\emptyset^{(n)} \leq_T \mathcal{W}_y^{\emptyset^{(n)}}$, then $(\mathcal{W}_{g(y)})^{(n)} \equiv_T \mathcal{W}_y^{\emptyset^{(n)}}$. Then

$$x \in \emptyset^{(n+3)} \Leftrightarrow \mathcal{W}_{f(x)}^{\emptyset^{(n)}} \equiv_T \emptyset^{(n)} \Leftrightarrow (\mathcal{W}_{gf(x)})^{(n)} \equiv_T \emptyset^{(n)} \Leftrightarrow \mathcal{W}_{gf(x)} \in \mathbf{L}_n.)$$

b) $\{x : \mathcal{W}_x \in \mathbf{H}_n\}$ *is* Σ^0_{n+4}*-complete.* (Schwarz [199?]) (Hint: similar, using X.9.16.)

c) $\{x : \mathcal{W}_x \in \mathbf{I}\}$ *is* $\Pi^0_{\omega+1}$*-complete.* (Solovay) (Hint: by the uniformity of parts a) and b), the index set of $\bigcup_{n\in\omega}(\mathbf{L}_n \cup \mathbf{H}_n)$ is r.e. in $\emptyset^{(\omega)}$. Conversely, consider $\emptyset^{(\omega+1)}$, which is complete for sets r.e. in $\emptyset^{(\omega)}$. Let h be a recursive function such that

$$x \in \emptyset^{(\omega+1)} \Leftrightarrow (\exists n)[h(x,n) \in \emptyset^{(n)}],$$

and e be such that $\{e\}^{\emptyset^{(n)}}(0) \simeq n$, i.e. e tells appropriate oracles apart. Using Sacks Jump Inversion Theorem and the Fixed-Point Theorem (with parameters) as on p. 655 or in XI.1.22.c, there is a recursive function f such that:

$$\mathcal{W}^X_{f(x)} = \begin{cases} X & \text{if } \{e\}^X(0) \simeq n \ \wedge \ h(x,n) \in X \\ X <_T \mathcal{W}^X_{f(x)} <_T X' & \\ \wedge\, (\mathcal{W}^X_{f(x)})' \equiv_T \mathcal{W}^{X'}_{f(x)} & \text{if } \{e\}^X(0) \simeq n \ \wedge \ h(x,n) \notin X \\ \emptyset & \text{if } \{e\}^X(0)\uparrow. \end{cases}$$

Then

$$\begin{aligned} x \in \emptyset^{(\omega+1)} &\Rightarrow (\exists n)(\mathcal{W}^{\emptyset}_{f(x)} \in \mathbf{L}_n) \\ x \notin \emptyset^{(\omega+1)} &\Rightarrow \mathcal{W}^{\emptyset}_{f(x)} \in \mathbf{I}, \end{aligned}$$

and $\{x : \mathcal{W}_x \in \mathbf{I}\}$ is $\Pi^0_{\omega+1}$-complete.)

Exercises X.9.19 In X.9.12 and X.9.14 we have classified both $\{x : \mathcal{W}_x$ recursive$\}$ and $\{x : \mathcal{W}_x$ creative$\}$.

a) $\{x : \mathcal{W}_x$ *simple*$\}$ *and* $\{x : \mathcal{W}_x$ *hypersimple*$\}$ *are* Π^0_3*-complete.* (Rogers [1959], Mostowski, Ehrenfeucht) (Hint: from III.2.14 and III.3.13.)

b) $\{x : \mathcal{W}_x$ *hyperhypersimple*$\}$, $\{x : \mathcal{W}_x$ *r-maximal* $\}$ *and* $\{x : \mathcal{W}_x$ *maximal* $\}$ *are* Π^0_4*-complete.* (Yates [1966a], Lachlan, Martin, Robinson) (Hint: let A be Σ^0_4-complete. As in the proof of X.9.18.a, there is a recursive function f such that

$$\begin{aligned} x \in A &\Rightarrow \mathcal{W}_{f(x)} \in \mathbf{L}_1 \\ x \notin A &\Rightarrow \mathcal{W}_{f(x)} \in \mathbf{H}_1. \end{aligned}$$

If g is a recursive function, obtained from IX.2.24, such that $\mathcal{W}_{g(x)} \equiv_T \mathcal{W}_x$ and $\mathcal{W}_{g(x)}$ is maximal when $\mathcal{W}_x$ has high degree, then

$$\mathcal{W}_{f(x)} \in \mathbf{H}_1 \Leftrightarrow \mathcal{W}_{gf(x)} \text{ is maximal.)}$$

c) $\{x : \mathcal{W}_x$ *has no maximal superset* $\}$ *is* Π^0_5*-complete.* (Jockusch) (Hint: let A be Σ^0_5-complete. By X.9.18.a, there is a recursive function f such that

$$x \in A \ \Leftrightarrow\ \mathcal{W}_{f(x)} \in \mathbf{L}_2.$$

If g is a recursive function, obtained from IX.2.29, such that $\mathcal{W}_{g(x)} \equiv_T \mathcal{W}_x$, and $\mathcal{W}_{g(x)}$ has no maximal supersets when $\mathcal{W}_x$ has non-low$_2$ degree, then

$$\mathcal{W}_{f(x)} \in \mathbf{L}_2 \ \Leftrightarrow\ \mathcal{W}_{gf(x)} \text{ has a maximal superset}$$

by IX.2.29 and its stated converse.)

Lempp [1987] has proved that $\{x : \mathcal{W}_x$ *quasimaximal* $\}$ *is* Σ^0_5*-complete.*

For additional results on specific index sets see Kallibekov [1971a], Stob [1982], Lempp [1987], Schwarz [1989], [199?], Jockusch, Lerman, Soare and Solovay [1989].

Applications of index sets to degrees

Following Rogers [1959], we now give *two new solutions to Post's Problem* by using index sets.

1. Notice that

$$\{x : \mathcal{W}_x \text{ nonrecursive}\} \supseteq \{x : \mathcal{W}_x \ T\text{-complete}\}$$

   by definition. But the first set is Π^0_3 (by X.9.12), while the second is not (since the proof of X.9.14.a shows that every Σ^0_3 set is m-reducible to it). They must thus be different, i.e. there exists a nonrecursive and not T-complete r.e. set.

2. Since **Rec** is Σ^0_3, from the proof of X.9.14.a we obtain a recursive function f such that

$$\mathcal{W}_x \text{ recursive} \ \Leftrightarrow\ \mathcal{W}_{f(x)} \ T\text{-complete}.$$

   By the Fixed-Point Theorem, there is e such that $\mathcal{W}_e = \mathcal{W}_{f(e)}$. Then

$$\mathcal{W}_e \text{ recursive} \ \Leftrightarrow\ \mathcal{W}_e \ T\text{-complete},$$

   and thus $\mathcal{W}_e$ is neither recursive nor T-complete.

Of course these two solutions to Post's Problem are not particularly useful, since their proofs use the very same methods used to give a direct solution, although part 1 gives a stronger result, namely that the index sets of recursive and T-complete sets have different m-degree. But these ideas can be pushed further with genuine advantage, as we now see.

As a second example of applications of index sets to degrees, we show how to obtain from X.9.17.a *a new proof of the existence of an r.e. degree incomparable to a given nontrivial r.e. degree* (X.6.1). Let C be an r.e. nonrecursive and T-incomplete set. Since $\{x : \mathcal{W}_x \leq_T C\}$ is Σ_4^0 (being $\Sigma_3^{0,C}$), there is a recursive function f such that

$$\begin{aligned}\mathcal{W}_x \leq_T C \quad &\Rightarrow \quad \mathcal{W}_{f(x)} \equiv_T \mathcal{K}\\ \mathcal{W}_x \not\leq_T C \quad &\Rightarrow \quad C \not\leq_T \mathcal{W}_{f(x)}.\end{aligned}$$

By the Fixed-Point Theorem, there exists an e such that $\mathcal{W}_e = \mathcal{W}_{f(e)}$. By the first condition, if $\mathcal{W}_e \leq_T C$, then $\mathcal{W}_e \equiv_T \mathcal{K}$, contradicting the fact that $C <_T \mathcal{K}$. Thus $\mathcal{W}_e \not\leq_T C$ and, by the second condition, $C \not\leq_T \mathcal{W}_e$. Thus C and $\mathcal{W}_e$ are Turing incomparable.

Notice that this proof is uniform, but it does not use the coding method. Rather, the required diagonalization is indirectly carried out by the Fixed-Point Theorem.

As a final application we give *a new proof of the Density Theorem*, based on the next result.

Proposition X.9.20 (Yates [1966a]) *Given r.e. sets $D <_T C$ and $A \in \Sigma_3^{0,C}$, there exists a recursive function f such that*

$$\begin{gathered}D \leq_T \mathcal{W}_{f(x)} \leq_T C\\ x \in A \Leftrightarrow \mathcal{W}_{f(x)} \equiv_T C.\end{gathered}$$

Proof. Let $\mathcal{W}_{g(x)}$ be as in X.9.15, and consider B_x defined as

$$\begin{aligned}\langle 0, z\rangle \in B_x \quad &\Leftrightarrow \quad z \in D\\ \langle e+1, z\rangle \in B_x \quad &\Leftrightarrow \quad \langle e, z\rangle \in \mathcal{W}_{g(x)}.\end{aligned}$$

Then $D \leq_T B_x \leq_T C$. Modify the proof of X.9.15 to obtain $\mathcal{W}_{f(x)} \subseteq B_x$ such that

$$\begin{aligned}x \in \overline{A} \quad &\Rightarrow \quad C \not\leq_T \mathcal{W}_{f(x)}\\ x \in A \quad &\Rightarrow \quad \mathcal{W}_{f(x)} \equiv_T C,\end{aligned}$$

the first condition using $C \not\leq_T D$, the second using $D \leq_T C$. □

Let now D and C be two r.e. sets such that $D <_T C$. We apply the result above with $A = \{x : \mathcal{W}_x \equiv_T D\}$, which is $\Sigma_3^{0,D}$ and hence $\Sigma_3^{0,C}$. Then

$$\mathcal{W}_x \equiv_T D \Leftrightarrow \mathcal{W}_{f(x)} \equiv_T C.$$

By the Fixed-Point Theorem, there exists an e such that $\mathcal{W}_e = \mathcal{W}_{f(e)}$. Then

$$\mathcal{W}_e \equiv_T D \Leftrightarrow \mathcal{W}_e \equiv_T C.$$

Since $D <_T C$, it follows that $\mathcal{W}_e \not\equiv_T D$ and $\mathcal{W}_e \not\equiv_T C$. But $D \leq_T \mathcal{W}_e \leq_T C$, so $D <_T \mathcal{W}_e <_T C$.

Global structure

We turn now to a global study of index sets.

Definition X.9.21 $\boldsymbol{\mathcal{I}_m}$ *is the substructure of* $\boldsymbol{\mathcal{D}_m}$ *consisting of the* m*-degrees of index sets.*

$\boldsymbol{\mathcal{I}_m}$ is closed under the operation $\boldsymbol{a} \mapsto \overline{\boldsymbol{a}}$ induced by complementation, since if $\mathcal{A}$ is a family of r.e. sets, then so is $\overline{\mathcal{A}}$, and $\theta\overline{\mathcal{A}} = \overline{\theta\mathcal{A}}$. If $\boldsymbol{a} \in \boldsymbol{\mathcal{I}_m}$, then $\boldsymbol{a}$ and $\overline{\boldsymbol{a}}$ are incomparable. Indeed, if there is a recursive function f such that

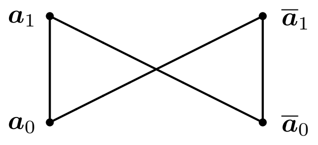
$$\mathcal{W}_x \in \mathcal{A} \Leftrightarrow \mathcal{W}_{f(x)} \in \overline{\mathcal{A}}$$

then, by the Fixed-Point Theorem, there is e such that $\mathcal{W}_e = \mathcal{W}_{f(e)}$. Thus

$$\mathcal{W}_e \in \mathcal{A} \Leftrightarrow \mathcal{W}_e \in \overline{\mathcal{A}},$$

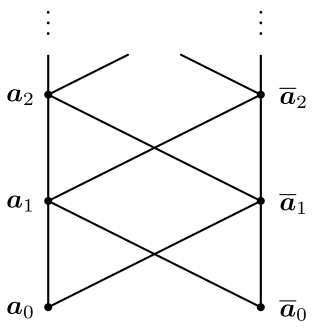
which is a contradiction.

$\boldsymbol{\mathcal{I}_m}$ has two minimal elements, namely $\boldsymbol{a}_0 = \{\emptyset\}$ and $\overline{\boldsymbol{a}}_0 = \{\omega\}$: they are the m-degrees of the index sets of trivial families. If $\boldsymbol{c}$ is the m-degree of the index set of a nontrivial family then, by the proof of X.9.3, $\boldsymbol{c}$ is above one of $\boldsymbol{a_1} = \mathbf{0}'_m$ (the m-degree of **Nem**) and $\overline{\boldsymbol{a}}_1$. We thus have the following initial segment:



This shows in particular that $\boldsymbol{\mathcal{I}_m}$ *is neither an upper nor a lower semilattice.*

More results on the local structure of $\boldsymbol{\mathcal{I}_m}$ are in Hay [1972], [1973], [1973a], [1974]. In particular, it is shown there that the m-degrees of index sets of finite families of r.e. sets (see X.9.7) and their complements have the following structure:



$\boldsymbol{a}_2$
$\overline{\boldsymbol{a}}_2$
$\boldsymbol{a}_1$
$\overline{\boldsymbol{a}}_1$
$\boldsymbol{a}_0$
$\overline{\boldsymbol{a}}_0$

Thus each of $\boldsymbol{a}_{n+1}$ and $\overline{\boldsymbol{a}}_{n+1}$ is the immediate successor of $\boldsymbol{a}_n$ and $\overline{\boldsymbol{a}}_n$, respectively, and every successor of both $\boldsymbol{a}_n$ and $\overline{\boldsymbol{a}}_n$ is above one of $\boldsymbol{a}_{n+1}$ and $\overline{\boldsymbol{a}}_{n+1}$ (this structure is *not* an initial segment of $\boldsymbol{\mathcal{I}_m}$). Thus $\boldsymbol{\mathcal{I}_m}$ fails quite badly to be an uppersemilattice. This example suggests the following definition.

Definition X.9.22 *A partially ordered structure* $(S, \sqsubseteq)$ *is a* **generalized uppersemilattice** *if, for every finite set* $X \subseteq S$*, there is a finite set* $Y \subseteq S$ *such that*

1. $(\forall x \in X)(\forall y \in Y)(x \sqsubseteq y)$
2. $(\forall z \in S)[(\forall x \in X)(x \sqsubseteq z) \Rightarrow (\exists y \in Y)(y \sqsubseteq z)]$.

The notion of **generalized lowersemilattice** *can be defined in a dual way.*

Given X, if Y_1 and Y_2 both satisfy the conditions above, then the set of minimal elements w.r.t. $\sqsubseteq$ of Y_1 and Y_2 coincide. We can thus talk of the **generalized l.u.b.** of X.

Obviously, if every pair of elements of S has a generalized l.u.b., then S is a generalized uppersemilattice.

Definition X.9.23 *A partially ordered structure* $(S, \sqsubseteq)$ *is a* **generalized uppersemilattice of rank** $\boldsymbol{n}$ *if every finite set* $X \subseteq S$ *has a generalized l.u.b. with cardinality at most* n*.*

The minimal rank measures in some sense how far a generalized uppersemilattice is from being an uppersemilattice, i.e. from having rank 1. The picture on p. 634 shows a generalized upper and lower semilattice of rank 2.

Theorem X.9.24 (Selivanov [1979]) $\boldsymbol{\mathcal{I}_m}$ *is a generalized uppersemilattice of rank 4.*

Proof. Given families $\mathcal{A}_1, \dots, \mathcal{A}_n$ of r.e. sets, we want to find families $\mathcal{B}_1, \dots, \mathcal{B}_4$ such that $\{\mathcal{B}_1, \dots, \mathcal{B}_4\}$ is the generalized l.u.b. of $\{\mathcal{A}_1, \dots, \mathcal{A}_n\}$. Suppose we have a family $\mathcal{A}$ of r.e. sets which is an upper bound of $\{\mathcal{A}_1, \dots, \mathcal{A}_n\}$. There are recursive functions g_i $(1 \leq i \leq n)$ such that

$$\mathcal{W}_x \in \mathcal{A}_i \iff \mathcal{W}_{g_i(x)} \in \mathcal{A}.$$

Split ω into n disjoint infinite recursive sets R_i, and fix $a_i \in R_i$. Consider recursive permutations $p_i : \omega \longrightarrow R_i - \{a_i\}$. We can mirror $\mathcal{A}_i$ into R_i in the following way:

$$\mathcal{A}_i^* = \{\{a_i\} \cup p_i(\mathcal{W}_x) : \mathcal{W}_x \in \mathcal{A}_i\}.$$

$\mathcal{A}_i^*$ is an isomorphic copy of $\mathcal{A}_i$, but with the following advantages:

1. the $\mathcal{A}_i^*$'s are disjoint (since $\mathcal{A}_i^*$ is a family of subsets of R_i);
2. if an r.e. set is in $\mathcal{A}_i^*$, then it has a_i as a member and it is contained in R_i.

We can try a reduction of $\mathcal{B} = \bigcup_{1 \leq i \leq n} \mathcal{A}_i^*$ to $\mathcal{A}$ by using the g_i's as follows. Let

$$\mathcal{W}_{f(x)} = \begin{cases} \emptyset & \text{if } (\forall i)(a_i \notin \mathcal{W}_x) \\ \mathcal{W}_{g_i h(x)} & \text{if } a_i \in \mathcal{W}_x \subseteq R_i \\ \omega & \text{otherwise,} \end{cases}$$

where $\mathcal{W}_{h(x)} = p_i^{-1}(\mathcal{W}_x - \{a_i\})$. First, note that f is well-defined. Indeed, we let $\mathcal{W}_{f(x)} = \emptyset$ until we discover that, for some i, $a_i \in \mathcal{W}_x$. Then we let $\mathcal{W}_{f(x)} = \mathcal{W}_{g_i h(x)}$ until we find an element in $\mathcal{W}_x$ which is not in R_i. If and when this happens, we let $\mathcal{W}_{f(x)} = \omega$.

By 2 above, if $a_i \notin \mathcal{W}_x$ for each i, or $\mathcal{W}_x \not\subseteq R_i$ for some i, then $\mathcal{W}_x$ is not in $\bigcup_{1 \leq i \leq n} \mathcal{A}_i^*$. If $a_i \in \mathcal{W}_x \subseteq R_i$, then i is unique and $\mathcal{W}_x$ could be in $\mathcal{A}_i^*$. We first go back from $\mathcal{W}_x$ (contained in R_i) to $\mathcal{W}_{h(x)}$ (the r.e. set of which $\mathcal{W}_x$ is the image in R_i). Then

$$\mathcal{W}_x \in \mathcal{A}_i^* \iff \mathcal{W}_{h(x)} \in \mathcal{A}_i \iff \mathcal{W}_{g_i h(x)} \in \mathcal{A}.$$

We now define four sets $\mathcal{B}_1, \dots, \mathcal{B}_4$ such that f m-reduces one of them to $\mathcal{A}$.

- If $\emptyset, \omega \notin \mathcal{A}$, then $\mathcal{W}_{f(x)} \notin \mathcal{B}$ in the first and third case of the definition of f. Thus $\mathcal{B} \leq_m \mathcal{A}$ via f and we can let $\mathcal{B}_1 = \mathcal{B}$.
- If $\emptyset \in \mathcal{A}$ and $\omega \notin \mathcal{A}$, then the third clause is still good but the first is not. However, if we let

  $$\mathcal{B}_2 = \mathcal{B} \cup \{\mathcal{W}_x : (\forall i)(a_i \notin \mathcal{W}_x)\},$$

  we have $\mathcal{B}_2 \leq_m \mathcal{A}$.

- If $\emptyset \notin \mathcal{A}$ and $\omega \in \mathcal{A}$, we can similarly let
$$\mathcal{B}_3 = \mathcal{B} \cup \{\mathcal{W}_x : (\exists i)(a_i \in \mathcal{W}_x \not\subseteq R_i)\}.$$
- If $\emptyset, \omega \in \mathcal{A}$, then let
$$\mathcal{B}_4 = \mathcal{B} \cup \{\mathcal{W}_x : (\forall i)(a_i \notin \mathcal{W}_x)\} \cup \{\mathcal{W}_x : (\exists i)(a_i \in \mathcal{W}_x \not\subseteq R_i)\}.$$

It is clear that the function t_i such that
$$\mathcal{W}_{t_i(x)} = \{a_i\} \cup p_i(\mathcal{W}_x),$$
reduces $\mathcal{A}_i$ to any one of $\mathcal{B}_1, \ldots, \mathcal{B}_4$. Thus $\{\mathcal{B}_1, \ldots, \mathcal{B}_4\}$ is the generalized l.u.b. of $\{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$. □

While the previous result showed a resemblance of $\boldsymbol{\mathcal{I}_m}$ to an uppersemilattice, the next result exposes a difference.

Proposition X.9.25 (Selivanov [1979]) *In $\boldsymbol{\mathcal{I}_m}$ no pair of incomparable elements has l.u.b.*

Proof. Given $\mathcal{A}_1$ and $\mathcal{A}_2$, let $\mathcal{A}$ be their l.u.b. We show that $\mathcal{A} \leq_m \mathcal{A}_1$ or $\mathcal{A} \leq_m \mathcal{A}_2$, so that $\mathcal{A}_1$ and $\mathcal{A}_2$ are comparable.

Consider the $\mathcal{B}_1, \ldots, \mathcal{B}_4$ of the previous result (for the given $\mathcal{A}_1$, $\mathcal{A}_2$). Since they are an upper bound of $\mathcal{A}_1$ and $\mathcal{A}_2$, while $\mathcal{A}$ is their l.u.b., $\mathcal{A} \leq_m \mathcal{B}_j$.

First, we want a recursive function g such that, in the notation of the previous proof,
$$\mathcal{W}_x \in \mathcal{A} \Leftrightarrow \mathcal{W}_{g(x)} \in \bigcup_{i=1,2} \mathcal{A}_i^*$$
and
$$(\forall x)(\exists i)(a_i \in \mathcal{W}_{g(x)} \subseteq R_i).$$
We fix j, $1 \leq j \leq 4$ (chosen as stated below). Let
$$\mathcal{W}_x \in \mathcal{A} \Leftrightarrow \mathcal{W}_{f(x)} \in \mathcal{B}_j.$$
We simply have to avoid the two troublesome cases
$$(\forall i)(a_i \notin \mathcal{W}_{f(x)}) \qquad \text{and} \qquad (\exists i)(a_i \in \mathcal{W}_{f(x)} \not\subseteq R_i).$$
By the Fixed-Point Theorem, there is a recursive function t such that
$$\mathcal{W}_{t(x)} = \begin{cases} \emptyset & \text{if } (\forall i)(a_i \notin \mathcal{W}_{ft(x)}) \\ \mathcal{W}_x & \text{if } a_i \in \mathcal{W}_{ft(x)} \subseteq R_i \\ \omega & \text{otherwise.} \end{cases}$$

By hypothesis, we have

$$\mathcal{W}_{t(x)} \in \mathcal{A} \Leftrightarrow \mathcal{W}_{ft(x)} \in \mathcal{B}_j.$$

To avoid $(\forall i)(a_i \notin \mathcal{W}_{ft(x)})$, which would produce $\mathcal{W}_{t(x)} = \emptyset$, we require:

- $j = 1$ or 3 if $\emptyset \in \mathcal{A}$ (so $\mathcal{W}_{ft(x)} \notin \mathcal{B}_j$)
- $j = 2$ or 4 if $\emptyset \notin \mathcal{A}$ (so $\mathcal{W}_{ft(x)} \in \mathcal{B}_j$).

To avoid $(\exists i)(a_i \in \mathcal{W}_{ft(x)} \not\subseteq R_i)$ we similarly require:

- $j = 1$ or 2 if $\omega \in \mathcal{A}$
- $j = 3$ or 4 if $\omega \notin \mathcal{A}$.

Thus, if j is the unique choice determined by whether $\emptyset$ and ω are in $\mathcal{A}$ or not, we have $\mathcal{W}_{t(x)} = \mathcal{W}_x$ for every x, and so

$$\mathcal{W}_x \in \mathcal{A} \Leftrightarrow \mathcal{W}_{t(x)} \in \mathcal{A} \Leftrightarrow \mathcal{W}_{ft(x)} \in \mathcal{B}_j.$$

We can thus let $g = ft$.

We now want to use g to obtain a reduction of $\mathcal{A}$ to one of $\mathcal{A}_1$, $\mathcal{A}_2$. It is enough to reduce $\mathcal{A}$ to one of $\mathcal{A}_1^*$ and $\mathcal{A}_2^*$. Note that, since R_1 and R_2 are disjoint and $(\forall x)(\exists i)(a_i \in \mathcal{W}_{g(x)} \subseteq R_i)$, for every x either $a_1 \in \mathcal{W}_{g(x)}$ or $a_2 \in \mathcal{W}_{g(x)}$, but not both. The obstacle in obtaining the required reduction is that we might have:

- $\mathcal{W}_x \in \mathcal{A}$ and $\mathcal{W}_{g(x)} \in \mathcal{A}_1^*$, for infinitely many x
- $\mathcal{W}_x \in \mathcal{A}$ and $\mathcal{W}_{g(x)} \in \mathcal{A}_2^*$, for infinitely many x.

We look directly at the conditions $a_1 \in \mathcal{W}_{g(x)}$ and $a_2 \in \mathcal{W}_{g(x)}$ and define, by the Fixed-Point Theorem, a recursive function s such that

$$\mathcal{W}_{s(x)} = \mathcal{W}_n \text{ if } a_i \in \mathcal{W}_{gs(x)} \text{ and } x \text{ is the } n\text{-th } z \text{ such that } a_i \in \mathcal{W}_{gs(z)}.$$

For the least such i, $\{x : a_i \in \mathcal{W}_{gs(x)}\}$ is infinite and recursive. If h is an enumeration of it in increasing order, then

$$\mathcal{W}_x \in \mathcal{A} \Leftrightarrow \mathcal{W}_{sh(x)} \in \mathcal{A} \Leftrightarrow \mathcal{W}_{gsh(x)} \in \mathcal{A}_i^*,$$

since $a_i \in \mathcal{W}_{gsh(x)}$ for every x. $\square$

This result shows that in $\boldsymbol{\mathcal{I}_m}$, the generalized l.u.b. of a pair of elements always has at least two elements. Hay [1972] shows that in many cases two elements are indeed enough (see the figure on p. 634). Selivanov [1979] has proved that this is not always the case, and that even 3 elements are not always enough. In particular, *$\mathcal{I}_m$ is not a generalized uppersemilattice of rank 3*.

Kusmina [1981] has proved that *the structure of m-degrees of Kleene index sets is a generalized uppersemilattice of rank 2 which is not an uppersemilattice.*

Exercises X.9.26 (Selivanov [1979]) a) *If Y is the generalized l.u.b. of X in $\mathcal{I}_m$, then X is the generalized g.l.b. of Y.* In particular, every finite set is the generalized g.l.b. of some finite set. (Hint: see the proof of X.9.25.)

b) *$\mathcal{I}_m$ is not a generalized lowersemilattice.* (Hint: let $\mathcal{A}$ be such that $\theta\mathcal{A}$ is Σ^0_2-complete. The generalized g.l.b. of $\theta\mathcal{A}$ and $\theta\overline{\mathcal{A}}$, let it be X, consists of elements m-reducible to them, and hence Δ^0_2. By part a), $\{\theta\mathcal{A}, \theta\overline{\mathcal{A}}\}$ is the generalized l.u.b. of X. By the proof of X.9.24, the generalized l.u.b. of Δ^0_2 sets consists of Δ^0_2 elements, which is a contradiction.)

æ

Chapter XI

Limit Sets

The subject of this chapter is the study of the degrees of Δ_2^0 sets or, equivalently, of the structure $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ of the degrees below $\mathbf{0}'$. The Limit Lemma (IV.1.17) shows that the Δ_2^0 sets are the sets that can be approximated by a recursive function, i.e. by an effective trial and error procedure, and this makes them a natural class to be investigated for their own sake. Another reason for interest in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ comes from the special role played in $\boldsymbol{\mathcal{D}}$ by $\mathbf{0}'$, which is simultaneously the greatest r.e. degree and the smallest jump.

We first concentrate on interesting subclasses of the Δ_2^0 sets, namely, **jump classes** in Section 1 and the class of **1-generic degrees** in Section 2. We then turn to the **Turing degrees** of Δ_2^0 sets and consider finite extension arguments in Section 3, tree arguments in Section 4, and global results in Section 5. In Section 6 we briefly consider the $\boldsymbol{m}$**-degrees** of Δ_2^0 sets.

XI.1 Jump Classes

The idea of this section is to stratify $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ by considering the iterated jump behavior of its elements. We consider only finite iterations, i.e. n-th jumps, since all arithmetical sets, and in particular all Δ_2^0 sets, have the same ω-jump (see XIII.1.11.a).

Domination properties

$\mathbf{0}'$ is not only the greatest degree in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$. It is also placed sufficiently high to be able to look down and see all the (partial) recursive functions in one glance. We give a precise formulation of this fact and consider degrees that share the same property with $\mathbf{0}'$.

Recall (III.3.9) that a function f dominates φ if, for almost every argument x, $f(x) \geq \varphi(x)$ whenever $\varphi(x)\downarrow$. We can easily define *a function $f \leq_T \mathcal{K}$ which dominates every partial recursive function*, e.g.

$$f(x) = 1 + \max\{\varphi_e(x) : e \leq x \wedge \varphi_e(x)\downarrow\}.$$

Exercise XI.1.1 *$\boldsymbol{a} \geq \mathbf{0}'$ if and only if there is a function recursive in $\boldsymbol{a}$ which dominates every partial recursive function.* (Tennenbaum [1961]) (Hint: see III.3.9.d. More directly, consider $g(x) \simeq \mu s.(x \in \mathcal{K}_s)$.)

Proposition XI.1.2 (Martin [1966a]) *The following are equivalent:*

1. *$\boldsymbol{a}' \geq \mathbf{0}''$*
2. *there is a function recursive in $\boldsymbol{a}$ which dominates every total recursive function.*

Proof. Suppose $\boldsymbol{a}' \geq \mathbf{0}''$. We want to dominate every total recursive function, recursively in $\boldsymbol{a}$. The condition 'φ_e is total' is Π^0_2 and hence recursive in $\mathbf{0}''$ and $\boldsymbol{a}'$. By relativization of the Limit Lemma (IV.1.17), there is g recursive in $\boldsymbol{a}$ such that

$$\begin{aligned} \varphi_e \text{ total} \quad &\Rightarrow \quad \lim_{s\to\infty} g(e,s) = 1 \\ \varphi_e \text{ not total} \quad &\Rightarrow \quad \lim_{s\to\infty} g(e,s) = 0. \end{aligned}$$

Thus, for given e and x, either $\varphi_e(x)\downarrow$ or we can find $s \geq x$ such that $g(e,s) = 0$. Define $f(x)$ as 1 plus the maximum of $\varphi_e(x)$ for all $e \leq x$ such that $\varphi_e(x)$ converges before we find $s \geq x$ such that $g(e,s) = 0$. Then f is recursive in $\boldsymbol{a}$ because g is. If φ_e is total, then f dominates it, because $g(e,s) = 1$ from a certain point s_e on, and for $x \geq s_e$ we have $f(x) > \varphi_e(x)$.

Now suppose that $f \leq_T \boldsymbol{a}$ dominates every total recursive function. Since the predicate 'φ_e is total' is Π^0_2-complete and hence of degree $\mathbf{0}''$, the Limit Lemma allows us to show that $\mathbf{0}'' \leq \boldsymbol{a}'$ by finding $g \leq_T \boldsymbol{a}$ such that

$$\begin{aligned} \varphi_e \text{ total} \quad &\Rightarrow \quad \lim_{s\to\infty} g(e,s) = 1 \\ \varphi_e \text{ not total} \quad &\Rightarrow \quad \lim_{s\to\infty} g(e,s) = 0. \end{aligned}$$

Let

$$g(e,s) = \begin{cases} 1 & \text{if } (\forall x \leq s)(\varphi_{e,f(s)}(x)\downarrow) \\ 0 & \text{otherwise.} \end{cases}$$

Then g is recursive in $\boldsymbol{a}$ because f is. If φ_e is total, then the function

$$h(s) = \mu z.\,(\forall x \leq s)(\varphi_{e,z}(x)\downarrow)$$

is total recursive, hence it is dominated by f. Then, for some s_e,

$$s \geq s_e \;\Rightarrow\; f(s) \geq h(s),$$

and

$$s \geq s_e \;\Rightarrow\; g(e,s) = 1.$$

If φ_e is not total, then there is x such that $\varphi_e(x)$ does not converge, and hence

$$s \geq x \;\Rightarrow\; g(e,s) = 0. \quad \Box$$

With no additional effort, we obtain a characterization of another class of degrees.

Proposition XI.1.3 *The following are equivalent:*

1. $\boldsymbol{b}'' = (\boldsymbol{b} \cup \mathbf{0}')'$
2. *there is a function recursive in* $\boldsymbol{b} \cup \mathbf{0}'$ *which dominates every total function recursive in* $\boldsymbol{b}$.

Proof. The relativization of XI.1.2 to $\boldsymbol{b}$ shows that $(\boldsymbol{b} \cup \boldsymbol{a})' \geq \boldsymbol{b}''$ if and only if there is a function recursive in $\boldsymbol{b} \cup \boldsymbol{a}$ which dominates every function recursive in $\boldsymbol{b}$. By letting $\boldsymbol{a} = \mathbf{0}'$, the first condition becomes $(\boldsymbol{b} \cup \mathbf{0}')' \geq \boldsymbol{b}''$, and the converse $(\boldsymbol{b} \cup \mathbf{0}')' \leq \boldsymbol{b}''$ is always true. $\Box$

Exercises XI.1.4 a) *The total recursive functions and the recursive sets are uniformly recursive in* $\mathbf{0}'$. (Hint: let

$$f(e,x) = \begin{cases} \varphi_e(x) & \text{if } (\forall z \leq x)(\varphi_e(z)\downarrow) \\ 0 & \text{otherwise.} \end{cases}$$

Similarly for sets, using characteristic functions.)

b) $\boldsymbol{a}' \geq \mathbf{0}''$ *if and only if the total recursive functions are uniformly recursive in* $\boldsymbol{a}$. (Jockusch [1972b]) (Hint: if $\boldsymbol{a}' \geq \mathbf{0}''$, there is g recursive in $\boldsymbol{a}$ which dominates all recursive functions. Let

$$f(\langle e,m\rangle, x) = \begin{cases} \varphi_e(x) & \text{if } (\forall z \leq x)(\varphi_{e,m+g(z)}(z)\downarrow) \\ 0 & \text{otherwise.} \end{cases}$$

The reason for the presence of m is that if h is total recursive, then $h(z) \leq g(z)$ only for all sufficiently large z, but $h(z) \leq m + g(z)$ everywhere, for an appropriate m. This ensures that if φ_e is total, then it is enumerated by f. Conversely, if the total recursive functions are uniformly enumerated by f recursive in $\boldsymbol{a}$, then $\max_{e \leq x} f(e,x)$ is recursive in $\boldsymbol{a}$ and dominates all recursive functions.)

c) $\boldsymbol{a}' \geq \mathbf{0}''$ *if and only if the recursive sets are uniformly recursive in* $\boldsymbol{a}$. (Jockusch [1972b]) (Hint: using a fixed 0,1-valued partial recursive function without total recursive extensions (II.2.2.a), define g recursive such that $\varphi_{g(e)}$ is 0,1-valued and

$$\begin{array}{rcl} \varphi_e \text{ total} & \Rightarrow & \varphi_{g(e)} \text{ total} \\ \varphi_e \text{ not total} & \Rightarrow & \varphi_{g(e)} \text{ has no total recursive extension.} \end{array}$$

Now let $f \leq_T \boldsymbol{a}$ enumerate the recursive sets. Then φ_e is total if and only if $\varphi_{g(e)}$ has one of the functions enumerated by f as an extension. Thus 'φ_e is total' is Σ^0_2 in $\boldsymbol{a}$. Since it is always Π^0_2, it is recursive in $\boldsymbol{a}'$.)

d) $\boldsymbol{a}'' = (\boldsymbol{a} \cup \mathbf{0}')'$ *if and only if the sets recursive in* $\boldsymbol{a}$ *are uniformly recursive in* $\boldsymbol{a} \cup \mathbf{0}'$. (Jockusch [1972b]) (Hint: by relativization of part c).)

By XI.1.3, a function recursive in $\mathbf{0}'$ can dominate every function recursive in $\boldsymbol{a} < \mathbf{0}'$ only if $\boldsymbol{a}'' = \mathbf{0}''$. In particular, the existence of degrees $\boldsymbol{a} < \mathbf{0}'$ such that $\boldsymbol{a}'' > \mathbf{0}''$ (XI.1.11) shows that there is no function recursive in $\mathbf{0}'$ that dominates every function recursive in any $\boldsymbol{a} < \mathbf{0}'$. The next result shows that something weaker can be achieved.

Proposition XI.1.5 (Tennenbaum [1961], Miller and Martin [1968], Yates [1967]) *There is a function recursive in* $\mathbf{0}'$ *not dominated by any function recursive in any* $\boldsymbol{a} < \mathbf{0}'$.

Proof. By the Limit Lemma, there is a recursive function g such that

$$\mathcal{K}(x) = \lim_{s \to \infty} g(x, s).$$

If

$$m_{\mathcal{K}}(x) = \mu s.\, (\forall t \geq s)[g(x,t) = g(x,s)],$$

then $m_{\mathcal{K}}$ is recursive in $\mathcal{K}$. And, for every x, $m_{\mathcal{K}}$ produces the first stage in which $g(x,s)$ has stabilized. If $m_{\mathcal{K}}(x) \leq f(x)$, then

$$\mathcal{K}(x) = g(x, f(x)).$$

Thus if f dominates $m_{\mathcal{K}}$, then $\mathcal{K} \leq_T f$. In particular, f cannot be recursive in any $A <_T \mathcal{K}$. $\square$

Corollary XI.1.6 (Posner [1981a]) *If S is infinite and recursive in* $\mathbf{0}'$ *then there is a function recursive in* $\mathbf{0}'$ *not dominated on S by any function recursive in any* $\boldsymbol{a} < \mathbf{0}'$.

Proof. Let

$$f(x) \quad = \quad \max_{z \leq x} m_{\mathcal{K}}(z),$$

where $m_{\mathcal{K}}$ is as in XI.1.5. Then f is not dominated by any function recursive in any $\boldsymbol{a} < \mathbf{0}'$, and moreover it is nondecreasing. Given S, consider the step function obtained by considering only the values of f on elements of S:

$$g(x) \quad = \quad f(\mu y.[y \in S \wedge y \geq x]).$$

Clearly g is recursive in $\mathbf{0}'$, because so are f and S, and $g(x) \geq f(x)$ because f is nondecreasing. If there is a function recursive in $\boldsymbol{a} < \mathbf{0}'$ that dominates g on S, then there is a nondecreasing function that dominates f on ω, contradicting the choice of f. $\square$

Given a set $A(x) = \lim_{s\to\infty} g(x,s)$ with g recursive, there are two functions that monitor stages in which g reads correctly:

$$\begin{aligned} m_A(x) &\quad = \quad \mu s.\, (\forall t \geq s)[g(x,t) = g(x,s)] \\ C_A(x) &\quad = \quad \mu s.\, [g(x,s) = A(x)]. \end{aligned}$$

The first is called the **modulus** of g, and it provides the first place after which g never reads wrongly. The second is called the **computation function** of g, and it provides the first place at which g reads correctly. Obviously A is recursive in both, since

$$A(x) \;=\; g(x, m_A(x)) \;=\; g(x, C_A(x)).$$

But while C_A is always recursive in A (by definition), in general m_A is only recursive in $\mathcal{K}$, due to the presence of the unbounded quantifier in its definition.

Exercise XI.1.7 *It is possible to choose g in such a way that m_A is recursive in A if and only if A has r.e. degree.* (Shoenfield [1959]) (Hint: if A is r.e., then the natural g changes at most once, and only if an element enters A. Conversely, consider the graph G of m_A. By definition of modulus, G is co-r.e. Since $A \leq_T m_A$, A is recursive in G. If $m_A \leq_T A$, then $G \leq_T A$ and $A \equiv_T G$.)

The next exercises are a version of V.5.3.d, and show again that every nonzero degree below $\mathbf{0}'$ is hyperimmune. They are proved as in XI.1.5, by use of the modulus function in the case of r.e. degrees, and by (an extension of) the computation function in the general case.

Exercises XI.1.8 a) *If A is r.e. and nonrecursive, there is a function recursive in A not dominated by any recursive function.* (Dekker [1954]) (Hint: for the natural g, consider m_A. Since $A(x) = g(x, m_A(x))$, if f dominates m_A, then A is recursive in f. Moreover, since A is r.e., $m_A \leq_T A$.)

b) *If $A \leq_T \mathcal{K}$ is nonrecursive, there is a function recursive in A not dominated by any recursive function.* (Miller and Martin [1968]) (Hint: consider the version of the computation function of A defined by

$$C_A(x) \ = \ \mu s.\, (s > C_A(x-1) \wedge (\forall y \leq x)[g(y,s) = A(y)]),$$

which is recursive in A. Since C_A is increasing, $C_A(x) \geq x$. If f dominates C_A, then $f(n) \geq C_A(n) \geq n$. To compute $A(x)$ from f, look for an interval $[n, f(n)]$ such that $n \geq x$ and $g(x,t)$ is independent of t on the interval. One of the t's in the interval must be $C_A(n)$, and then

$$g(x, C_A(n)) = A(x)$$

by the definition of C_A.)

Jump classes

The results proved in the previous subsection show that interesting properties of a degree are related to its jump, thus suggesting a classification of degrees according to the value of their jump.

Definition XI.1.9 The Jump Hierarchy (Sacks [1963], Cooper [1974], Soare [1974]) *For any n:*

1. *$\mathbf{L}_n$ is the class of degrees $\boldsymbol{a} \leq \mathbf{0}'$ such that $\boldsymbol{a}^{(n)} = \mathbf{0}^{(n)}$, and such degrees are called* $\mathbf{low}_n$
2. *$\mathbf{H}_n$ is the class of degrees $\boldsymbol{a} \leq \mathbf{0}'$ such that $\boldsymbol{a}^{(n)} = \mathbf{0}^{(n+1)}$, and such degrees are called* $\mathbf{high}_n$
3. *$\mathbf{I}$ is the class of degrees $\boldsymbol{a} \leq \mathbf{0}'$ such that $(\forall n)(\mathbf{0}^{(n)} < \boldsymbol{a} < \mathbf{0}^{(n+1)})$, and such degrees are called* **intermediate**.

Thus low$_n$ and high$_n$ degrees are those with the smallest and the greatest possible value of the n-th jump, respectively. In particular, low$_1$ and high$_1$ degrees, called simply **low** and **high** degrees, have the smallest and greatest possible jump.

By convention $\boldsymbol{a}^{(0)} = \boldsymbol{a}$, so that $\mathbf{L}_0 = \{\mathbf{0}\}$ and $\mathbf{H}_0 = \{\mathbf{0}'\}$. XI.1.2 and XI.1.3 thus give the following characterizations of the high and low$_2$ degrees, respectively.

Corollary XI.1.10 Characterization of $\mathbf{H}_1$ and $\mathbf{L}_2$

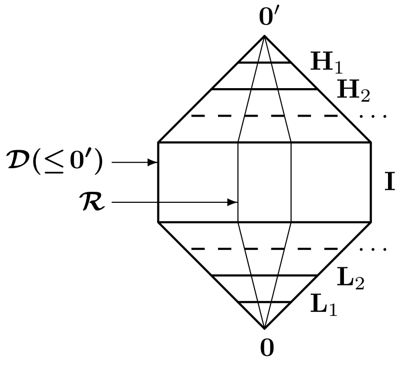
1. *A degree $\boldsymbol{a} \leq \mathbf{0}'$ is high if and only if there is a function recursive in $\boldsymbol{a}$ which dominates every total recursive function.*
2. *A degree $\boldsymbol{a} \leq \mathbf{0}'$ is low$_2$ if and only if there is a function recursive in $\mathbf{0}'$ which dominates every function recursive in $\boldsymbol{a}$.*

Proof. If $\boldsymbol{a} \leq \mathbf{0}'$, then automatically $\boldsymbol{a}' \leq \mathbf{0}''$ and $\boldsymbol{a} \cup \mathbf{0}' = \mathbf{0}'$. $\square$

The properties of the jump operator (in particular monotonicity, see V.1.14) produce the following trivial properties of jump classes:

1. $\{\mathbf{L}_n\}_{n\in\omega}$ *and* $\{\mathbf{H}_n\}_{n\in\omega}$ *are ascending sequences of classes of degrees*
2. *each jump class* $\mathbf{L}_n$ *and* $\mathbf{H}_n$ *is, respectively, downward and upward closed w.r.t. the ordering of degrees below* $\mathbf{0}'$
3. $(\bigcup_{n\in\omega} \mathbf{L}_n) \cap (\bigcup_{n\in\omega} \mathbf{H}_n) = \emptyset$.

We thus have an exhaustive hierarchy for $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ that can be illustrated as follows:



It remains to prove that the hierarchy is proper, i.e. that *none of the classes* $\mathbf{L}_{n+1} - \mathbf{L}_n$, $\mathbf{H}_{n+1} - \mathbf{H}_n$ *and* $\mathbf{I}$ *is empty*. We now consider the case $n = 0$, while the general case is discussed in the next subsection.

We already know (from the proof of V.2.21) that there exists a nontrivial low degree, i.e. a degree $\boldsymbol{a}$ such that $\boldsymbol{a} > \mathbf{0}$ and $\boldsymbol{a}' = \mathbf{0}'$, and thus $\mathbf{L}_1 - \mathbf{L}_0 \neq \emptyset$. Since the proof was a finite extension argument, no such argument can produce degrees in $\overline{\mathbf{L}_1}$ for the usual reasons (see Section V.3). We thus build a nontrivial high degree by coinfinite extensions.

Proposition XI.1.11 (Sacks [1963b]) *There exists a degree* $\boldsymbol{a}$ *such that* $\boldsymbol{a} < \mathbf{0}'$ *and* $\boldsymbol{a}' = \mathbf{0}''$*, and thus* $\mathbf{H}_1 - \mathbf{H}_0 \neq \emptyset$*.*

Proof. Let $C \in \mathbf{0}''$. We want to build $A <_T \mathcal{K}$ such that $A' \equiv_T C$. The construction will be recursive in $\mathcal{K}$, and thus we will automatically have $A \leq_T \mathcal{K}$ and hence $A' \leq_T \mathcal{K}' \equiv_T C$. It remains to ensure that $\mathcal{K} \not\leq_T A$ and $C \leq_T A'$.

To satisfy $C \leq_T A'$, we will make sure that

$$\begin{aligned} x \in C \quad &\Rightarrow \quad \lim_{s\to\infty} A(\langle x, s\rangle) = 1 \\ x \notin C \quad &\Rightarrow \quad \lim_{s\to\infty} A(\langle x, s\rangle) = 0. \end{aligned}$$

Then $C \leq_T A'$ by the Limit Lemma. Unlike what we did in V.4.3, we cannot simply code $C(x)$ in the x-th column of A at a chosen stage, because C is not recursive in $\mathcal{K}$, and thus we cannot simply compute $C(x)$ if we want to keep the construction recursive in $\mathcal{K}$. We have two options:

1. We may take C r.e. in $\mathcal{K}$ (which is possible because $\mathbf{0''}$ is r.e. in $\mathbf{0'}$) and choose a one-one enumeration f of C recursive in $\mathcal{K}$. When a new element x is generated in C, we put the x-th column in A, with at most finitely many exceptions.

2. Since $C \leq_T \mathcal{K}'$, by the Limit Lemma we can approximate $C(x)$ by a function $C_s(x)$ uniformly recursive in $\mathcal{K}$, and work at any stage with the approximations.

To satisfy $\mathcal{K} \not\leq_T A$, we have the requirements

$$R_e \quad : \quad \mathcal{K} \not\simeq \{e\}^A$$

and we look for e-splittings (compatible with what has been done so far) in the usual way. We also want to avoid having $\lim_{s\to\infty} A(\langle x, s\rangle) = 1$ if $x \notin C$. Now it might happen that, to satisfy the requirements R_e, we keep on committing elements of the x-th column (for a fixed $x \notin C$) by putting them into A. To avoid this we let the condition R_e interfere with the x-th column (by putting elements into it) only if $e \leq x$, so that only finitely many elements of the x-th column can enter A for the sake of such requirements.

We give a proof following the first suggestion (see XI.1.12.a for a proof following the second one). Let f be a one-one enumeration of C recursive in $\mathcal{K}$. We start with $\theta_0 = \emptyset$. At stage $s+1$, given θ_s, we look for the least $e \leq s$ such that R_e has not yet been satisfied, and there are e-splitting strings σ_1 and σ_2 compatible with θ_s, and giving only value 0 to elements of the columns with index smaller than e (if θ_s is not already defined on them).

If there is such an e, let σ_1 and σ_2 be the smallest strings as above and choose the σ_i that produces a disagreement with $\mathcal{K}$ on the smallest element on which they e-split. We then define

$$\theta_{s+1}(x) \simeq \begin{cases} \theta_s(x) & \text{if } \theta_s(x)\downarrow \\ \sigma_i(x) & \text{if } \sigma_i(x)\downarrow \\ 1 & \text{if } x = \langle f(s), z\rangle, \text{ otherwise} \\ 0 & \text{if } x = \langle n, z\rangle \ \wedge \ n \neq f(s) \ \wedge \ n, z \leq s. \end{cases}$$

If there is no such e, we define θ_{s+1} as above with $\sigma_i = \emptyset$. Let $A = \bigcup_{s\in\omega} \theta_s$.

The last clause in the definition of θ_{s+1} ensures that A is total (since columns corresponding to elements of $\overline{C}$ are only finitely affected by the rest of the construction). By construction, $A \leq_T \mathcal{K}$ and

$$\begin{aligned} x \in C \quad &\Rightarrow \quad \lim_{s\to\infty} A(\langle x, s\rangle) = 1 \\ x \notin C \quad &\Rightarrow \quad \lim_{s\to\infty} A(\langle x, s\rangle) = 0. \end{aligned}$$

It remains to prove by induction on e that R_e has been satisfied. Choose s large enough so that the requirements R_i with $i < e$ have been satisfied before stage s, if ever, and such that the elements smaller than e which are in C have been generated before stage s. If R_e has been satisfied before stage s, there is nothing to prove. Otherwise, if there are e-splittings at stage $s+1$ as required, then R_e is satisfied at stage $s+1$ by the choice of θ_{s+1}. If they do not exist, consider

$$\theta_s^*(x) = \begin{cases} \theta_s(x) & \text{if } \theta_s(x)\downarrow \\ 0 & \text{if } x = \langle n, z\rangle \ \wedge\ n < e, \text{ otherwise.} \end{cases}$$

θ_s^* gives the final value of A on the first e columns, because by hypothesis on s no action is taken after stage $s+1$ to satisfy requirements R_i for $i < e$. Then, for any x, any string σ compatible with θ_s^* and making $\{e\}^\sigma(x)$ converge must produce a unique value. This gives a procedure recursive in θ_s^* to compute the function $\{e\}^A$, if total. But θ_s^* can be finitely described, since it consists of a finite string and of the first e columns (whose value is eventually constant) and it is thus recursive. Thus $\{e\}^A$ is recursive if total, hence different from $\mathcal{K}$, and R_e is satisfied. $\Box$

Exercises XI.1.12 a) *Give an alternative proof of XI.1.11, following the second suggestion given above.* (Hint: this uses only finite strings. At stage $s+1$, given σ_s consider, for each $e \leq s$,

$$\theta_s^e(x) \simeq \begin{cases} \sigma_s(x) & \text{if } \sigma_s(x)\downarrow \\ C_s(n) & \text{if } x = \langle n, z\rangle \ \wedge\ n < e, \text{ otherwise,} \end{cases}$$

and see if there are e-splittings compatible with it. Note that one cannot simply proceed as in V.4.3, and use coinfinite conditions defined (except for finitely many values) as

$$\theta_s(x) \simeq C_z(n) \text{ if } x = \langle n, z\rangle \ \wedge\ n < e,$$

because questions about the existence of e-splittings compatible with them would only be r.e. in $\mathbf{0}'$, and not recursive in it.)

b) *There is a minimal pair of high degrees.* (Lachlan [1966b]) (Hint: combine XI.1.11 and V.2.16, by building A and B simultaneously. Note that one cannot simply build A given B of high degree, because questions on whether $\{e\}^B(x)$ converges would not automatically be recursive in $\mathbf{0}'$, unless B is low.)

Exercises XI.1.13 High r.e. degrees. (Sacks [1963], [1963b])

a) *There is an incomplete high r.e. degree.* (Hint: given C r.e. in $\mathbf{0}'$ of degree $\mathbf{0}''$, using the fact that C is the range of a function that can be recursively approximated, or that $C \in \Sigma_2^0$, define B r.e. such that

$$x \in C \ \Rightarrow\ B^{[x]} \text{ finite} \quad \text{and} \quad x \notin C \ \Rightarrow\ B^{[x]} = \omega,$$

so that $C \leq_T B'$. Then apply the Thickness Lemma X.3.7 to B, which is piecewise recursive, to obtain an r.e. thick subset A of B (so that $C \leq_T A'$) such that $\mathcal{K} \not\leq_T A$.)

b) *If C is r.e. nonrecursive there is $A <_T C$ such that $A' \equiv_T C'$.* (Hint: define an r.e. set B recursive in C such that

$$\{e\}^C(e)\!\downarrow\ \Rightarrow\ B^{[e]} \text{ finite} \qquad \{e\}^C(e)\!\uparrow\ \Rightarrow\ B^{[e]} = \omega,$$

so that $C' \leq_T B'$. Then apply the Strong Thickness Lemma X.3.9 to obtain an r.e. thick subset A of B such that $A \leq_T B$ (so that $A \leq_T C$) and $C \not\leq_T A$.)

c) *Every incomplete degree below $\mathbf{0}'$ is bounded by an incomplete high degree.* (Hint: relativize part b).)

d) *Every incomplete r.e. degree is bounded by an incomplete r.e. high degree.* (Hint: given an r.e. set $C <_T \mathcal{K}$, obtain as in part b) an r.e. set B whose 0-th column codes C, and such that

$$\{e\}^{\mathcal{K}}(e)\!\downarrow\ \Rightarrow\ B^{[e+1]} \text{ finite} \qquad \{e\}^{\mathcal{K}}(e)\!\uparrow\ \Rightarrow\ B^{[e+1]} = \omega,$$

so that $C \leq_T B$ and $\mathcal{K}' \leq_T B'$. Then apply the Thickness Lemma to B, which is piecewise recursive in $C <_T \mathcal{K}$, to obtain an r.e. thick subset A of B such that $\mathcal{K} \not\leq_T A$.)

Hops

The existence proof of a high degree in XI.1.11 is methodologically simple, since it uses only coinfinite extensions and the cancellation method, i.e. priority without injuries. The proof has been constructivized in XI.1.13 to obtain high r.e. degrees, this time using infinite injury (in the codified form of the Thickness Lemma).

We now give a different proof of XI.1.11, which uses finite injury, because it refers to the existence of a nontrivial r.e. low degree. The interest in this proof lies in the fact that a natural constructivization of it produces a finite injury construction of a high r.e. degree, thus improving on the exercises above. More importantly, having r.e. low and high degrees one can iterate the method to show, with no additional effort and still by finite injury, that no level of the Jump Hierarchy collapses.

The main idea is the following observation. Given a set A, the relativization to A of the construction of an r.e. set $\mathcal{W}_i$ corresponds naturally to the set $A \oplus \mathcal{W}_i^A$. If we can show that for any i there is a set A such that $A \oplus \mathcal{W}_i^A \equiv_T \mathcal{K}$, then we have turned the construction of any r.e. degree with a certain property to a construction of a degree such that $\mathbf{0}'$ has the same property relative to it. Since a high degree is just a degree relative to which $\mathbf{0}'$ is low, we can pass from a low r.e. degree to a high degree.

The operators involved in this construction are going to be useful in other contexts, and we thus introduce names and notation for them.

Definition XI.1.14 (Jockusch and Shore [1983]) *The operator*

$$X \longmapsto X \oplus \mathcal{W}_i^X$$

is called a **hop***, and $X \oplus \mathcal{W}_i^X$ is denoted by* $\boldsymbol{H_i(X)}$*.*

The reason for the name 'hop' is the fact that one has

$$X \leq_T H_i(X) \leq_T X',$$

i.e. a hop always goes up in degree, possibly not strictly, and it can be at most a jump. We will see that some of the properties of the jump generalize to any hop. As a first example, the next result generalizes the fact that there is a set with jump $\mathcal{K}$, i.e. of low degree (V.2.21).

Proposition XI.1.15 (Jockusch and Shore [1983]) *For any i, there is A such that $H_i(A) \equiv_T \mathcal{K}$.*

Proof. The proof is similar to that of V.2.24, except that in place of deciding whether $\{e\}^A(e)\downarrow$, i.e. whether $e \in \mathcal{W}_e^A$, we decide whether $e \in \mathcal{W}_i^A$.

As usual we build A by finite initial segments, starting with $\sigma_0 = \emptyset$. At stage $s+1$, let σ_s be given.

- If $s = 2e$, then we see if there is $\sigma \supseteq \sigma_s$ such that $e \in \mathcal{W}_i^\sigma$. If so, we let $\sigma_{s+1} = \sigma$ for the smallest such σ, and otherwise we let $\sigma_{s+1} = \sigma_s$.
- If $s = 2e+1$, we code the e-th element of $\mathcal{K}$ into A:

$$\sigma_{s+1} = \sigma_s * \langle \mathcal{K}(e) \rangle.$$

The construction is recursive in $\mathcal{K}$, since the first step asks an r.e. question, while the second uses $\mathcal{K}$ directly. Thus $A \leq_T \mathcal{K}$. Moreover, $\mathcal{W}_i^A \leq_T \mathcal{K}$ because

$$e \in \mathcal{W}_i^A \;\Leftrightarrow\; e \in \mathcal{W}_i^{\sigma_{2e+1}},$$

and thus $H_i(A) = A \oplus \mathcal{W}_i^A \leq_T \mathcal{K}$.

The construction is also recursive in $\mathcal{W}_i^A$ and A, as follows. Knowing $\mathcal{W}_i^A$ we can decide whether $e \in \mathcal{W}_i^A$, hence if the string σ required at step $2e+1$

exists. If so, we just look recursively for it until we find the smallest such string. Step $2e+2$ simply determines the value of A for the next undefined element, which is $|\sigma_{2e+1}|$. Then

$$e \in \mathcal{K} \Leftrightarrow \sigma_{2e+2}(|\sigma_{2e+1}|) = 1,$$

and $\mathcal{K} \leq_T A \oplus \mathcal{W}_i^A = H_i(A)$. $\square$

Exercise XI.1.16 Hops Inversion Theorem. *For any hop H_i, if $\mathcal{K} \leq_T C$ then there is A such that $H_i(A) \equiv_T A \oplus \mathcal{K} \equiv_T C$.* (Jockusch and Shore [1983]) (Hint: as above, but argue separately that $H_i(A) \equiv_T C$ and $A \oplus \mathcal{K} \equiv_T C$, since in general it is not true that $A \oplus \mathcal{K} \leq_T H_i(A)$, as it was in V.2.24 for the jump.)

An immediate consequence of XI.1.15 is the following new proof of existence of a nontrivial high degree. Consider an r.e. nonrecursive set $\mathcal{W}_i$ which is low, i.e. such that

$$\emptyset <_T \mathcal{W}_i \ \wedge \ (\mathcal{W}_i)' \equiv_T \mathcal{K}.$$

The usual constructions of such an r.e. set uniformly relativize, i.e. they actually produce i such that, for every X,

$$X <_T \mathcal{W}_i^X \ \wedge \ (\mathcal{W}_i^X)' \equiv_T X'.$$

Let A be as in XI.1.15, for such an i. Then

$$A \ <_T \ \mathcal{W}_i^A \ \equiv_T A \oplus \mathcal{W}_i^A \ \equiv_T \ \mathcal{K},$$

so $A <_T \mathcal{K}$ and $A' \equiv_T (\mathcal{W}_i^A)' \equiv_T \mathcal{K}'$, and A has nontrivial high degree.

One can immediately guess that, as high degrees can be obtained from r.e. low degrees, low$_2$ degrees could be obtained from r.e. high degrees in a similar way, by applying XI.1.15 to an i such that $\mathcal{W}_i^X$ is a nontrivial high degree, uniformly in X. We would obviously like to start an inductive process by using as i the solution we have just obtained, but the problem is that XI.1.15 produces a set A which is not necessarily r.e. We thus need to strengthen the result in the following way.

Proposition XI.1.17 (Jockusch and Shore [1983]) *For any i, there is a nonrecursive r.e. set A such that $H_i(A) \equiv_T \mathcal{K}$.*

Proof. This introduces the natural modifications in the proof of XI.1.15 required to make A r.e. and nonrecursive. The requirements are

$$\begin{array}{lll} R_{3e} & : & A \not\simeq \{e\} \\ R_{3e+1} & : & \text{decide } e \in \mathcal{W}_i^A \\ R_{3e+2} & : & \text{if } e \in \mathcal{K} \text{ then leave a trace in } A. \end{array}$$

The strategies for the single requirements are the natural ones:

- For R_{3e}, pick a fresh witness $x_e^s \notin A_s$ and wait until $\{e\}(x_e^s)\downarrow$. If and when it does, put x_e^s into A if and only $\{e\}(x_e^s) \simeq 0$.
- For R_{3e+1}, wait until $e \in \mathcal{W}_i^{A_s}$, and then freeze the elements used negatively in the computation.
- For R_{3e+2}, pick a fresh witness $z_e^s \notin A_s$ and wait until e appears in $\mathcal{K}$. If and when it does, put z_e^s into A.

The construction of A is the standard one following the strategies described, and produces A r.e. In particular, $A \leq_T \mathcal{K}$.

Since the construction is recursive, we have a recursive function $f(e,s)$ which tells whether $e \in \mathcal{W}_i^{A_s}$. But then the limit of f is recursive in $\mathcal{K}$, and tells whether $e \in \mathcal{W}_i^A$. Indeed, after R_{3e+1} takes highest priority either $e \notin \mathcal{W}_i^{A_s}$ for every s, hence $e \notin \mathcal{W}_i^A$, or $e \in \mathcal{W}_i^{A_s}$ for some s and the situation is permanently frozen, hence $e \in \mathcal{W}_i^A$. Thus $\mathcal{W}_i^A \leq_T \mathcal{K}$, and $H_i(A) \leq_T \mathcal{K}$.

Finally, $\mathcal{K} \leq_T H_i(A)$ much as in XI.1.15. The oracles A and $\mathcal{W}_i^A$ allow us to check the construction and to find out whether a temporary situation is going to be final or not. Precisely, we know whether $e \in \mathcal{W}_i^A$, so we can wait until we find s such that $e \in \mathcal{W}_i^{A_s}$, and then recursively in A we can see if this is going to be injured or not, i.e. if some of the elements used negatively in the computation are going to enter A. This way we can find, given e and recursively in $H_i(A)$, the final witness z_e. Then $e \in \mathcal{K}$ if and only if $z_e \in A$. Thus $\mathcal{K} \leq_T H_i(A)$. $\square$

For our purposes we need the result in the following uniform strong form.

Corollary XI.1.18 Decomposition of a jump into two hops. *There is a recursive function f such that, for any choice H_i of a second hop, $H_{f(i)}$ is a nontrivial first hop that decomposes the jump. More precisely,*

$$X <_T H_{f(i)}(X) \quad \textit{and} \quad H_i(H_{f(i)}(X)) \equiv_T X'.$$

$$\begin{tikzcd}
H_i(H_{f(i)}(X)) \equiv_T X' \\
H_{f(i)}(X) \arrow[u, "H_i"] \\
X \arrow[u, "H_{f(i)}"]
\end{tikzcd}$$

Proof. The work described above shows that, for any hop H_i, there is an r.e.

nonrecursive set A such that $H_i(A) \equiv_T \emptyset' = \mathcal{K}$. Since the r.e. set A is the result of a hop H_e starting from the empty set, we have

$$(\forall i)(\exists e)[H_i(H_e(\emptyset)) \equiv_T \emptyset'].$$

The result then follows by relativization to X, and from the observation that e is obtained uniformly from i and hence, by the S^m_n-Theorem, via a recursive function of i. $\square$

Theorem XI.1.19 Hierarchy Theorem (Sacks [1963b], Lachlan [1965c], Martin [1966b]) *The Jump Hierarchy does not collapse. More precisely, for any n, the classes $\mathbf{L}_{n+1} - \mathbf{L}_n$, $\mathbf{H}_{n+1} - \mathbf{H}_n$ and $\mathbf{I}$ are nonempty. Actually, they all contain r.e. degrees.*

Proof. Let f be as XI.1.18. Then

$$\begin{array}{lcl}(\forall B)[(H_i(B))^{(n)} \equiv_T B^{(n)}] & \Rightarrow & (\forall C)[C^{(n+1)} \equiv_T (H_{f(i)}(C))^{(n)}] \\ (\forall B)[(H_i(B))^{(n)} \equiv_T B^{(n+1)}] & \Rightarrow & (\forall C)[C^{(n+1)} \equiv_T (H_{f(i)}(C))^{(n+1)}].\end{array}$$

Indeed, since $H_i(H_{f(i)}(X)) \equiv_T X'$ we have, for $B = H_{f(i)}(C)$,

$$C^{(n+1)} \equiv_T (C')^{(n)} \equiv_T (H_i(H_{f(i)}(C))^{(n)} \equiv_T (H_{f(i)}(C))^{(n)}$$

in the first case, and

$$C^{(n+1)} \equiv_T (C')^{(n)} \equiv_T (H_i(H_{f(i)}(C))^{(n)} \equiv_T (H_{f(i)}(C))^{(n+1)}$$

in the second. Thus f is a function that switches from low$_n$ to high$_n$, and from high$_n$ to low$_{n+1}$. In particular, starting from i such that

$$(\forall B)[H_i(B) \equiv_T B'],$$

i.e. from the index of a relativized low r.e. set, the sets

$$\emptyset, H_i(\emptyset), H_{f(i)}(\emptyset), H_{f(f(i))}(\emptyset), \ldots$$

successively visit all jump classes except $\mathbf{I}$. Moreover, if the solution given by i is nontrivial, i.e.

$$B <_T H_i(B),$$

then we actually visit all the classes $\mathbf{L}_{n+1} - \mathbf{L}_n$ and $\mathbf{H}_{n+1} - \mathbf{H}_n$.

The idea to obtain an intermediate degree is as follows. We would like an e such that

$$X <_T H_e(X) <_T X' \quad \text{and} \quad H_e(X') \equiv_T (H_e(X))',$$

i.e. a nontrivial hop that commutes with the jump operator. Then by iteration

$$\emptyset^{(n)} <_T H_e(\emptyset^{(n)}) = (H_e(\emptyset))^{(n)} <_T \emptyset^{(n+1)}.$$

The condition

$$H_e(X') \equiv_T (H_e(X))'$$

can be rewritten, using the fact that f decomposes the jump, as

$$H_e(H_e H_{f(e)}(X)) \equiv_T H_e H_{f(e)}(H_e(X)),$$

and it is obviously satisfied if e is a fixed-point of f, i.e. if $H_e = H_{f(e)}$. In this case the first condition of nontriviality is also satisfied, since then

$$X' \equiv_T H_e(H_{f(e)}(X)) = H_e(H_e(X)),$$

i.e. H_e decomposes the jump, and thus

$$X <_T H_e(X) <_T X' = H_e(H_e(X)).$$

Thus any hop corresponding to a fixed-point of f yields an r.e. set of intermediate degree, when applied to $\emptyset$. $\square$

The original proofs of the existence of intermediate degrees, by Lachlan [1965c] and Martin [1966b], were direct constructions based on modifications of the Sacks Jump Inversion Theorem (XI.1.23) combined with fixed-point arguments. Sacks [1966] obtained the same result by noting that one only needed a nontrivial relativized r.e. set commuting with the jump operator, i.e.

$$X <_T \mathcal{W}_e^X <_T X' \quad \text{and} \quad \mathcal{W}_e^{X'} = (\mathcal{W}_e^X)',$$

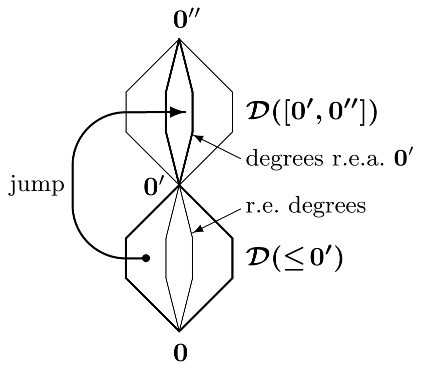
and that the existence of e followed without any additional argument, simply by the Fixed-Point Theorem, from the fact that the Sacks Jump Inversion Theorem holds in a uniform and relativized way.

All the proofs just quoted relied on (modifications of) the Sacks Jump Inversion Theorem and thus required infinite injury. Then Jockusch and Shore [1983] produced the proof given in XI.1.19 by exploiting the uniform relativization of XI.1.17, which requires only finite injury.

From a proof-theoretical point of view, i.e. in the setting of Bounded Arithmetic (see X.4.5 and the comments following it), *the existence of incomplete high r.e. degrees cannot be proved in First-Order Arithmetic with Σ_1-induction.* In particular, the proof of XI.1.19 uses the existence of nontrivial low r.e. degrees in a relativized form, and it is thus provable only in First-Order Arithmetic with parametrized Σ_1-induction, which is stronger than Σ_1-induction. By relativization, *the existence of intermediate r.e. degrees cannot be proved in First-Order Arithmetic with full arithmetical induction* (Mytilinaios and Slaman [1988]).

Jump inversion below $\mathbf{0}'$

We now determine the range of the jump operator restricted to $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$, to obtain the analogue of the Jump Inversion Theorem V.2.24. Note that for any degree $\boldsymbol{a}$, $\mathbf{0}' \leq \boldsymbol{a}'$ and $\boldsymbol{a}'$ is r.e. in $\boldsymbol{a}$. Moreover, if $\boldsymbol{a} \leq \mathbf{0}'$, then $\boldsymbol{a}'$ is r.e. in $\mathbf{0}'$. Thus the jump of a degree below $\mathbf{0}'$ must be r.e. in and above $\mathbf{0}'$. We now show that the condition is also sufficient.


$\mathbf{0}''$
$\boldsymbol{\mathcal{D}}([\mathbf{0}', \mathbf{0}''])$
degrees r.e.a. $\mathbf{0}'$
jump
$\mathbf{0}'$
r.e. degrees
$\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$
$\mathbf{0}$

Theorem XI.1.20 Shoenfield Jump Inversion Theorem (Shoenfield [1959]) *The range of the jump operator restricted to $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ is the set of degrees r.e. in and above $\mathbf{0}'$.*

Proof. Let C be r.e. in $\mathbf{0}'$ and such that $\mathcal{K} \leq_T C$. We want to build $A <_T \mathcal{K}$ such that $A' \equiv_T C$.

To satisfy $C \leq_T A'$, we will make sure that

$$\begin{aligned} x \in C \quad &\Rightarrow \quad \lim_{s\to\infty} A(\langle x, s\rangle) = 1 \\ x \notin C \quad &\Rightarrow \quad \lim_{s\to\infty} A(\langle x, s\rangle) = 0, \end{aligned}$$

so that $C \leq_T A'$ by the Limit Lemma. To ensure this we will follow the method of XI.1.11, by choosing a one-one enumeration f of C recursive in $\mathcal{K}$. When a new element x is generated in C, we put the x-th column in A, with at most finitely many exceptions.

To satisfy $A' \leq_T C$, we have the requirements

$$R_e \quad : \quad \text{decide whether } \{e\}^A(e)\downarrow,$$

to be satisfied recursively in C.

We build A by coinfinite extensions. We start with $\theta_0 = \emptyset$. At stage $s+1$, given θ_s we look for the least $e \leq s$ such that R_e has not yet been satisfied (i.e.

$\{e\}^{\theta_s}(e)\uparrow$), and there is a string σ compatible with θ_s and giving only value 0 to elements of the columns with index smaller than e (when θ_s is not defined on them), such that $\{e\}^{\sigma}(e)\downarrow$.

If there is such an e, let σ be the smallest string as above. We then define

$$\theta_{s+1}(x) \simeq \begin{cases} \theta_s(x) & \text{if } \theta_s(x)\downarrow \\ \sigma(x) & \text{if } \sigma(x)\downarrow \\ 1 & \text{if } x = \langle f(s), z\rangle, \text{ otherwise} \\ 0 & \text{if } x = \langle n, z\rangle \ \wedge\ n \neq f(s) \ \wedge\ n, z \leq s. \end{cases}$$

If there is no such e, we define θ_{s+1} as above, with $\sigma_i = \emptyset$. Let $A = \bigcup_{s\in\omega}\theta_s$.

As in XI.1.11 we have $A \leq_T \mathcal{K}$ and $C \leq_T A'$. It remains to prove by induction on e that R_e has been satisfied. Choose s_0 large enough so that the elements smaller than e which are in C have been generated before stage s_0. Note that s_0 can be found recursively in C, and that the first e-columns can later receive new elements only for the satisfaction of some R_i with $i < e$. If we let $s = s_0 + e$, then every R_i with $i < e$ that is satisfied during the construction, has been satisfied by stage s. For example, if R_0 can be satisfied at stage $s_0 + 1$, then it is, having highest priority; and if it cannot, then it never will. And similarly for each successive R_i, at each successive stage. Thus if R_e can be satisfied, it certainly is by stage $s+1$, i.e.

$$\{e\}^A(e)\downarrow \Leftrightarrow \{e\}^{\theta_{s+1}}(e)\downarrow .$$

Since s can be found recursively in C, and the construction is recursive in $\mathcal{K} \leq_T C$, the whole procedure is recursive in C and R_e is satisfied. $\square$

Corollary XI.1.21 (Spector [1956], Shoenfield [1959]) *Every possibility compatible with any of the properties*

$$\boldsymbol{a} \leq \boldsymbol{b} \ \Rightarrow\ \boldsymbol{a}' \leq \boldsymbol{b}' \quad \textit{and} \quad \boldsymbol{a}' \cup \boldsymbol{b}' \leq (\boldsymbol{a} \cup \boldsymbol{b})'$$

is realized in the degrees below $\mathbf{0}'$.

Proof. The proof of V.2.27 leaves only the following cases open:

1. $\boldsymbol{a}|\boldsymbol{b}$ *and* $\boldsymbol{a}' < \boldsymbol{b}'$
   Let $\boldsymbol{b}$ be a nontrivial high degree, i.e. $\boldsymbol{b} < \mathbf{0}'$ and $\boldsymbol{b}' = \mathbf{0}''$. Choose $\boldsymbol{a}$ low incomparable with it (see XI.3.5). Then $\boldsymbol{a}|\boldsymbol{b}$ and $\boldsymbol{a}' = \mathbf{0}' < \mathbf{0}'' = \boldsymbol{b}'$.

2. $\boldsymbol{a}|\boldsymbol{b}$ *and* $\boldsymbol{a}'|\boldsymbol{b}'$
   It is enough to take two incomparable degrees r.e. in and above $\mathbf{0}'$, which exist by the relativized Friedberg-Muchnick Theorem. The Shoenfield Jump Inversion Theorem produces two degrees below $\mathbf{0}'$ with (them as) incomparable jumps. $\square$

There is a certain asymmetry between domain and range in the statement of the Shoenfield Jump Inversion Theorem, since by considering *all* degrees below $\mathbf{0}'$ we obtain only the degrees *r.e.* in and above $\mathbf{0}'$. Historically, a symmetric version of the Jump Inversion Theorem replacing $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ by $\boldsymbol{\mathcal{R}}$ was needed not only for its own sake, but also to deduce the Hierarchy Theorem XI.1.19 (even just for degrees below $\mathbf{0}'$), as follows.

By relativizing the existence of a low nonrecursive r.e. degree to $\mathbf{0}'$ and applying XI.1.20, we obtain a degree below $\mathbf{0}'$ in $\mathbf{L}_2 - \mathbf{L}_1$, but we can only go so far. If we apply the same result relativized to $\mathbf{0}''$ instead of to $\mathbf{0}'$, we only obtain a degree between $\mathbf{0}'$ and $\mathbf{0}''$ with that jump (by XI.1.20 relativized). To further descend below $\mathbf{0}'$ and hence obtain a degree in $\mathbf{L}_3 - \mathbf{L}_2$, we would need a degree r.e. in $\mathbf{0}'$. A similar situation applies to the classes $\mathbf{H}_n$ once we know the existence of an *r.e.* incomplete high degree, since we can only obtain a degree in $\mathbf{H}_2 - \mathbf{H}_1$. Both these results give the existence of a degree $\boldsymbol{a} < \mathbf{0}'$ such that

$$\mathbf{0} < \boldsymbol{a} < \mathbf{0}' < \boldsymbol{a}' < \mathbf{0}'',$$

first proved by Friedberg [1957b], and again we have to stop here.

To strengthen XI.1.20 we can follow any of the two strategies adopted to strengthen XI.1.11: either a direct constructivization, requiring infinite injury (as in XI.1.13.a), or an indirect approach via hops, using only finite injury (as in XI.1.19). We sketch the first in the exercises, and describe the second in the next theorem.

Exercises XI.1.22 Jumps of r.e. degrees. a) *The range of the jump operator restricted to r.e. degrees is the set of degrees r.e. in and above* $\mathbf{0}'$. (Sacks [1963b]) (Hint: extend the proof of XI.1.13.a as in XI.1.20.)

b) *For every n, the classes* $\mathbf{L}_{n+1} - \mathbf{L}_n$ *and* $\mathbf{H}_{n+1} - \mathbf{H}_n$ *contain r.e. degrees.* (Sacks [1963b]) (Hint: use part a) relativized, and the existence of nontrivial low and high r.e. degrees.)

c) *The class* $\mathbf{I}$ *contains r.e. degrees.* (Lachlan [1965c], Martin [1966b]) (Hint: by a strengthening of part a), there is f recursive such that, for every X,

$$X <_T \mathcal{W}^X_{f(e)} <_T X' \quad \text{and} \quad (\mathcal{W}^X_{f(e)})' \equiv_T \mathcal{W}^{X'}_e .$$

Then use the Fixed-Point Theorem.)

d) *Every possibility compatible with any of the properties*

$$\boldsymbol{a} \leq \boldsymbol{b} \;\Rightarrow\; \boldsymbol{a}' \leq \boldsymbol{b}' \quad \textit{and} \quad \boldsymbol{a}' \cup \boldsymbol{b}' \leq (\boldsymbol{a} \cup \boldsymbol{b})'$$

is realized in the r.e. degrees. (Sacks [1963]) (Hint: As in XI.1.21, using part a) in place of XI.1.20.)

Simpson [1985] has noted that the same result XI.1.17 that allowed us to prove the Hierarchy Theorem for the jump classes (together with the Fixed-Point Theorem) also proves the Jump Inversion Theorem for r.e. degrees.

Theorem XI.1.23 Sacks Jump Inversion Theorem (Sacks [1963b]) *The range of the jump operator restricted to the r.e. degrees is the set of degrees r.e. in and above* $\mathbf{0}'$.

Proof. The basic idea is as follows. Given C r.e. in and above $\emptyset'$, C is the result of a hop H_e applied to $\emptyset'$. But $\emptyset'$ is a jump, hence the result of two hops by XI.1.18. Thus C is obtained by a sequence of three hops. By taking the first hop, which is an r.e. set, we arrive at C by two more hops, hence by a jump.

Recall the recursive function f of XI.1.18, that decomposes the jump into two hops:

$$H_i(H_{f(i)}(X)) \equiv_T X'.$$

If $C = H_e(\emptyset')$, then

$$\begin{array}{rcl} H_e(H_{f(e)}(H_{f(f(e))}(\emptyset))) & \equiv_T & (H_{f(f(e))}(\emptyset))' \\ H_{f(e)}(H_{f(f(e))}(\emptyset)) & \equiv_T & \emptyset' \end{array}$$

by definition of f. If H_e were degree-invariant, from this we would obtain

$$H_e(\emptyset') \equiv_T (H_{f(f(e))}(\emptyset))',$$

and $H_{f(f(e))}(\emptyset)$ would be a nontrivial r.e. set with jump C.

We cannot expect H_e to be degree-invariant in general, but the above reasoning works for any i. Since the second condition gives

$$H_{f(i)}(H_{f(f(i))}(\emptyset)) \equiv_T \emptyset'$$

for any i, and the proof of XI.1.18 actually exhibits a Turing reduction, there is a recursive function g such that, for all i,

$$H_{g(i)}(H_{f(i)}(H_{f(f(i))}(\emptyset))) \equiv_T H_e(\emptyset').$$

In other words, $H_e(\emptyset')$ is r.e. in and above $\emptyset'$, and hence r.e. in and above $H_{f(i)}(H_{f(f(i))}(\emptyset))$, and $H_{g(i)}$ witnesses this:

$$\begin{tikzcd}
C = H_e(\emptyset') \equiv_T H_{g(i)}(H_{f(i)}(H_{f(f(i))}(\emptyset))) & \\
\emptyset' \arrow[u, "H_e"] \arrow[r, dash, "\equiv_T"'] & H_{f(i)}(H_{f(f(i))}(\emptyset)) \arrow[ul, "H_{g(i)}"'] \\
& {} \arrow[u, "H_{f(i)}"] \\
& \emptyset \arrow[u, "H_{f(f(i))}"]
\end{tikzcd}$$

But now the Fixed-Point Theorem provides us with an i such that

$$H_i(X) = H_{g(i)}(X).$$

Then

$$H_i(H_{f(i)}(H_{f(f(i))}(\emptyset))) \equiv_T H_e(\emptyset'),$$

and by the first condition of f above

$$(H_{f(f(i))}(\emptyset))' \equiv_T H_e(\emptyset') = C.$$

Thus $H_{f(f(i))}(\emptyset)$ is a nontrivial r.e. set with jump C. $\Box$

Extensions of the Sacks and Shoenfield Jump Inversion Theorems have been considered in Robinson [1971a] and Shore [1988a].

Coles, Downey and LaForte [199?] have proved that *the Sacks Jump Inversion Theorem fails for wtt-reducibility*. On p. I.586 we proved that the Friedberg Jump Inversion Theorem (V.2.24) holds instead for any reducibility between $\leq_{tt}$ and $\leq_T$.

Generalized jump classes $\star$

A way to extend the Jump Hierarchy from $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ to $\boldsymbol{\mathcal{D}}$ is suggested by XI.1.3, and consists of replacing $\mathbf{0}'$ by $\boldsymbol{a} \cup \mathbf{0}'$.

Definition XI.1.24 The Generalized Jump Hierarchy (Jockusch [1977])

1. $\mathbf{GL}_n$ *is the class of degrees* $\boldsymbol{a}$ *such that* $\boldsymbol{a}^{(n)} = (\boldsymbol{a} \cup \mathbf{0}')^{(n-1)}$*, and such degrees are called* **generalized low**$_n$

2. $\mathbf{GH}_n$ *is the class of degrees* $\boldsymbol{a}$ *such that* $\boldsymbol{a}^{(n)} = (\boldsymbol{a} \cup \mathbf{0}')^{(n)}$*, and such degrees are called* **generalized high**$_n$

3. $\mathbf{GI}$ *is the class of degrees* $\boldsymbol{a}$ *which are neither generalized low*$_n$ *nor generalized high*$_n$*, for any* n*, and such degrees are called* **generalized intermediate**.

Notice that $\mathbf{GL}_1$ *is the set of degrees realizing the least possible jump*, i.e. of the degrees $\boldsymbol{a}$ such that $\boldsymbol{a}' = \boldsymbol{a} \cup \mathbf{0}'$, but an analogue fails at higher levels. For example, while *every degree realizing the least possible double jump is in* $\mathbf{GL}_2$ (because $\boldsymbol{a} \cup \mathbf{0}'' \leq (\boldsymbol{a} \cup \mathbf{0}')' \leq \boldsymbol{a}''$, and so if $\boldsymbol{a}'' = \boldsymbol{a} \cup \mathbf{0}''$, then $\boldsymbol{a}'' = (\boldsymbol{a} \cup \mathbf{0}')'$), *there are degrees in* $\mathbf{GL}_1$ *that do not realize the least possible double jump* (because if $\boldsymbol{a}' \geq \mathbf{0}''$, then $\boldsymbol{a} \cup \mathbf{0}'' \leq \boldsymbol{a}' \cup \mathbf{0}'' = \boldsymbol{a}' < \boldsymbol{a}''$, and by V.2.24 for any $\boldsymbol{c} \geq \mathbf{0}''$ there is $\boldsymbol{a}$ such that $\boldsymbol{a}' = \boldsymbol{a} \cup \mathbf{0}' = \boldsymbol{c}$).

The relationships between the two Jump Hierarchies are spelled out in the following result.

Proposition XI.1.25 *On* $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ *the generalized jump classes coincide with the jump classes, while on* $\boldsymbol{\mathcal{D}}$ *the former properly include the latter.*

Proof. The coincidence is trivial, since for $\boldsymbol{a} \leq \mathbf{0}'$ one has $\boldsymbol{a} \cup \mathbf{0}' = \mathbf{0}'$. For the nontrivial inclusions, pick any $\boldsymbol{a} > \mathbf{0}'$. First, $\boldsymbol{a}$ itself is in $\mathbf{GH_0}$, since $\boldsymbol{a} = \boldsymbol{a} \cup \mathbf{0}'$. Second, by V.2.24 there is $\boldsymbol{b}$ such that $\boldsymbol{b}' = \boldsymbol{b} \cup \mathbf{0}' = \boldsymbol{a}$, and $\boldsymbol{b}$ is in $\mathbf{GL_1}$ and not below $\mathbf{0}'$ (otherwise $\boldsymbol{b} \cup \mathbf{0}' = \mathbf{0}'$, while $\boldsymbol{b} \cup \mathbf{0}' = \boldsymbol{a} > \mathbf{0}'$). $\square$

The Hierarchy Theorem for the Jump Hierarchy thus also proves that the Generalized Jump Hierarchy does not collapse. However, it is difficult to picture the latter, since it lacks the closure properties held by the former. Indeed, if $\boldsymbol{a} \not\geq \mathbf{0}'$, then $\boldsymbol{a}$ has a successor in $\mathbf{GL}_1$ (XI.3.11.a). If we let $\boldsymbol{a}$ be a nontrivial high degree, then we have a degree in $\mathbf{GH}_1$ with a successor in $\mathbf{GL}_1$, which shows that *for any* n*,* $\mathbf{GL}_n$ *is not closed downwards and* $\mathbf{GH}_n$ *is not closed upwards.*

See Lerman [1985] for more on the ordering among degrees belonging to different (generalized) jump classes.

XI.2 1-Generic Degrees

The idea for the definition of 1-generic sets and degrees comes from the proofs of theorems about the jump operator in Chapter V. The sets A in those proofs all shared a basic property that allows control of their jump. Namely, one can determine on the basis of finite information about A, not only whether computations recursive in A converge (this is always true, by compactness of the recursive functionals, see II.3.13), but also whether they diverge. The basic tool used in those proofs is isolated in the following definition.

Definition XI.2.1 (Jockusch [1977]) *For given e and σ, $\{e\}^\sigma(x)$ is* **strongly undefined** *if $(\forall\tau \supseteq \sigma)(\{e\}^\tau(x)\uparrow)$, i.e. the computation diverges not only on σ, but also on every extension of it.*

A set A is **1-generic** *if, for every e and x, there is $\sigma \subseteq A$ such that $\{e\}^\sigma(x)$ is either defined or strongly undefined.*

A degree $\boldsymbol{a}$ is **1-generic** *if it contains a 1-generic set.*

Exercise XI.2.2 *A is 1-generic if and only if, for every e, there is $\sigma \subseteq A$ such that $\{e\}^\sigma(e)$ is either defined or strongly undefined.* (Hint: as in III.1.2)

The proofs of V.2.21, V.2.24 and V.2.26 all build 1-generic degrees with given additional properties. We now show that 1-generic degrees automatically have a number of the properties that can be realized by finite extension arguments recursive in $\emptyset'$. This accounts for the name '1-generic', as 'having all properties typical of the degrees below $\mathbf{0}'$'. More precision will be possible in Chapter XII, after the introduction of the machinery of forcing (see XII.1.14).

Proposition XI.2.3 (Jockusch [1977], Posner [1977], Jockusch and Posner [1978]) *Let $\boldsymbol{a}$ be a 1-generic degree. Then:*

1. *$\boldsymbol{a}$ is not recursive*
2. *$\boldsymbol{a} \in \mathbf{GL_1}$, in particular $\boldsymbol{a}$ is low if $\boldsymbol{a} \leq \mathbf{0}'$*
3. *$\boldsymbol{a}$ does not bound r.e. nonzero degrees, in particular $\boldsymbol{a}$ is not r.e.*
4. *$\boldsymbol{a}$ is the join of a minimal pair, in particular $\boldsymbol{a}$ is not minimal.*

Proof. To prove part 1, let A be 1-generic. If B is recursive, let e be such that

$$\{e\}^X(x) \simeq 1 \Leftrightarrow (\exists z)(B(z) \neq X(z)),$$

and $\{e\}^X(x)\uparrow$ otherwise. Suppose $\{e\}^A(x)\uparrow$. By 1-genericity

$$(\exists\sigma \subseteq A)(\forall\tau \supseteq \sigma)(\{e\}^\tau(x)\uparrow).$$

But this is impossible because, given σ, we can always find $X \supseteq \sigma$ such that $\{e\}^X(x)\downarrow$ (take any z on which σ is undefined and let X differ from B on it). So $\{e\}^A(x)\downarrow$ and $A \neq B$.

The proof of part 3 is similar. If B is r.e. and nonrecursive, we want to show $B \not\leq_T A$, i.e. $B \not\simeq \{i\}^A$ for any i. Since B is r.e., there is e such that

$$\{e\}^X(x) \simeq 1 \Leftrightarrow (\exists z)(z \in B \wedge \{i\}^X(z) \simeq 0).$$

Suppose $B \simeq \{i\}^A$. Then $\{e\}^A(x)\uparrow$, and by 1-genericity

$$(\exists \sigma \subseteq A)(\forall \tau \supseteq \sigma)(\{e\}^\tau(x)\uparrow).$$

Choose such a σ. Then B is recursive, which is a contradiction, because

$$z \in \overline{B} \Leftrightarrow (\exists \tau \supseteq \sigma)(\{i\}^\tau(z) \simeq 0).$$

The left-to-right implication follows because $B \simeq \{i\}^A$ by hypothesis. The right-to-left implication follows by definition of e, because no string τ extending σ makes $\{e\}^\tau(x)$ converge. Thus if $\{i\}^\tau(z) \simeq 0$, it cannot be that $z \in B$, and then $z \in \overline{B}$.

To prove part 2, note that

$$\begin{array}{rcl} e \in A' & \Leftrightarrow & (\exists \sigma)(\sigma \subseteq A \wedge \{e\}^\sigma(e)\downarrow) \\ e \in \overline{A'} & \Leftrightarrow & (\exists \sigma)(\sigma \subseteq A \wedge (\forall \tau \supseteq \sigma)(\{e\}^\tau(e)\uparrow)) \end{array}$$

by 1-genericity. Thus A' and $\overline{A'}$ are r.e. in $A \oplus \mathcal{K}$ and, by Post's Theorem II.1.14, $A' \leq_T A \oplus \mathcal{K}$. The reverse inequality always holds.

Finally, to prove part 4, let

$$X_0 = \{z : 2z \in X\} \quad \text{and} \quad X_1 = \{z : 2z+1 \in X\}.$$

Then $X = X_0 \oplus X_1$. Note that if A is 1-generic, then A_0 and A_1 are not recursive (they are even 1-generic), by an argument similar to that used above for A itself. We want to show that A_0 and A_1 form a minimal pair, i.e. that

$$\{i\}^{A_0} \simeq \{i\}^{A_1} \simeq C \Rightarrow C \text{ recursive.}$$

Let

$$\{e\}^X(x) \simeq 1 \Leftrightarrow (\exists z)[\{i\}^{X_j}(z)\downarrow (j=0,1) \wedge \{i\}^{X_0}(z) \not\simeq \{i\}^{X_1}(z)].$$

If $\{e\}^A(x)\downarrow$, then $\{i\}^{A_0} \not\simeq \{i\}^{A_1}$. If $\{e\}^A(x)\uparrow$, then, by 1-genericity,

$$(\exists \sigma \subseteq A)(\forall \tau \supseteq \sigma)(\{e\}^\tau(x)\uparrow).$$

Choose such a σ. If $\tau \supseteq \sigma$, then, by definition of e, $\{i\}^{\tau_0}(z)$ and $\{i\}^{\tau_1}(z)$ are equal if they both converge. If $\{i\}^{A_0} \simeq \{i\}^{A_1} \simeq C$, to compute $C(z)$ it is then enough to search for $\tau \supseteq \sigma$ such that $\{i\}^{\tau_0}(z)\downarrow$, and C is recursive. $\square$

Corollary XI.2.4 (Shoenfield [1959]) *There are degrees below* $\mathbf{0}'$ *not containing r.e. sets.*

Proof. The proof follows by part 3 above and the existence of 1-generic degrees below $\mathbf{0}'$. $\square$

Exercises XI.2.5 Let $\boldsymbol{a}$ be a 1-generic degree.

a) *Every countable partial ordering is embeddable in* $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$. (Jockusch [1977]) (Hint: see V.2.9.)

b) *The one-quantifier theory of* $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ *is independent of* $\boldsymbol{a}$. (Hint: see V.2.10.)

c) $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ *is not a lattice*. (Jockusch [1981]) (Hint: see V.4.9.b.)

Exercises XI.2.6 1-generic $\boldsymbol{m}$-degrees (Jockusch [1981]) Let $\boldsymbol{a}$ be a 1-generic m-degree.

a) $\boldsymbol{\mathcal{D}_m}(\leq \boldsymbol{a})$ *is isomorphic to* $\mathcal{E}^*$. (Hint: with notation as in VI.2.1, one only needs to show that Φ is one-one, i.e. that if B and C are r.e. sets differing infinitely, then $\Phi(B) \not\leq_m \Phi(C)$. Consider a possible recursive many-one reduction h. We want z such that

$$z \in \Phi(B) \Leftrightarrow h(z) \notin \Phi(C)$$

or, if B and C are the ranges of f and g,

$$f(z) \in A \Leftrightarrow g(h(z)) \notin A.$$

Let e be such that

$$\{e\}^X(x) \simeq 1 \Leftrightarrow (\exists z)(f(z) \in X \Leftrightarrow g(h(z)) \notin X).$$

Then $\{e\}^A(x)\downarrow$ because the ranges of f and g differ infinitely.)

b) *Every nonzero m-degree in* $\boldsymbol{\mathcal{D}_m}(\leq \boldsymbol{a})$ *is 1-generic*. (Hint: 1-genericity is preserved under inverse images by one-one recursive functions, and every infinite r.e. set B is the range of such a function. Moreover, every nonzero m-degree less than $\boldsymbol{a}$ is the m-degree of $\Phi(B)$, for some r.e. set B.)

Although, by XI.2.6.a, the structure $\boldsymbol{\mathcal{D}_m}(\leq \boldsymbol{a})$ of the m-degrees below a 1-generic m-degree $\boldsymbol{a}$ is unique up to isomorphism, *there are 1-generic degrees* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *such that* $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ *and* $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{b})$ *are not elementarily equivalent*. Indeed, some 1-generic degrees bound minimal degrees (Chong and Downey [1990], Kumabe [1990]) and some do not (Chong and Jockusch [1984], see XI.2.18). Moreover, Kumabe [199?] has proved that *there are 1-generic degrees* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *below* $\mathbf{0}'$ *such that* $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ *and* $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{b})$ *are not isomorphic*.

Full approximation arguments

The construction of sets recursive in $\emptyset'$ and degrees below $\mathbf{0}'$ can be done in two different ways:

1. by **$\mathbf{0}'$-oracle arguments**, in which the construction is recursive in $\mathcal{K}$ and we cannot change our mind once we have made a commitment
2. by **full approximation arguments**, in which the construction is recursive and we can change our mind finitely often for each element (the final set being recursive in $\mathcal{K}$ by the Limit Lemma).

We have already seen examples of both types of construction. $\mathbf{0}'$-oracle arguments were given in Chapter V, e.g. the construction of a 1-generic degree in V.2.21 was of this kind. Full approximation arguments were the standard procedure for the construction of r.e. sets, but there the full freedom of the method was not exploited, since the only change we allowed was from 0 to 1 (i.e. we could put new elements in the set, but not take them out). It is in the construction of degrees below $\mathbf{0}'$ that the full approximation method displays its full power.

We exemplify the method by giving a new construction of a 1-generic degree below $\mathbf{0}'$, which will turn out to be useful for generalizations.

Proposition XI.2.7 *1-generic degrees below $\mathbf{0}'$ can be built by full approximation.*

Proof. To make A 1-generic, we have the requirements

$$R_e \quad : \quad \{e\}^A(e)\downarrow \vee (\exists\sigma \subseteq A)(\forall\tau \supseteq \sigma)(\{e\}^\tau(e)\uparrow).$$

We first see what goes wrong if we try to build A by finite extensions σ_s, as usual, but by a recursive construction. We cannot ask whether there are strings that make certain computations converge, since this is an r.e. question. Thus, at each step, we only look for strings of a fixed length and wait until we find them.

We start with $\sigma_0 = \emptyset$. At stage $s+1$, given σ_s, we look for the least $e \leq s$ such that $\{e\}_s^{\sigma_s}(e)\uparrow$ and there exists $\sigma \supseteq \sigma_s$ with $|\sigma| \leq s$ and $\{e\}_s^\sigma(e)\downarrow$. If there is such an e, pick the smallest one, let σ be the smallest string as above, and let $\sigma_{s+1} = \sigma$. Otherwise, let $\sigma_{s+1} = \sigma_s$. To check that R_e is satisfied, we would like an s such that

$$\{e\}^{\sigma_s}(e)\downarrow \vee (\forall\sigma \supseteq \sigma_s)(\{e\}^\sigma(e)\uparrow).$$

The trouble is that at each stage s, we only look for strings extending σ_s. What might happen is that, for every s, there are strings $\sigma \supseteq \sigma_s$ such that $\{e\}^\sigma(e)\downarrow$, but with $|\sigma| > s$. To use them we thus have to wait until a later stage $t \geq |\sigma|$ is reached, but by then such strings might have become incompatible with σ_t.

The solution to this problem is the natural one. At stage $s+1$, we look not only for strings extending σ_s, but for any string. Suppose we are considering

R_0. Our first guess is that there is no string σ such that $\{0\}^\sigma(0)\downarrow$, and thus that R_0 is satisfied because

$$(\forall \sigma \supseteq \sigma_0)(\{0\}^\sigma(0)\uparrow).$$

If at a later stage $s+1$, we instead find a string σ such that $|\sigma| \leq s$ and $\{0\}^\sigma_s(0)\downarrow$, then we will let $\sigma_{s+1} = \sigma$, and again R_0 will be satisfied. Of course, σ might be incompatible with σ_s, and thus we will have to start all the work done for other requirements again, as in a typical priority argument. But from then on, we will remain above σ_{s+1}, since that will satisfy R_0.

The construction is as follows. We start with $\sigma_0 = \emptyset$. At stage $s+1$, given σ_s and $r(e,s)$ stating the amount of σ_s that we have to preserve to save the work done until now for R_e, let

$$R(e,s) = \max_{i<e} r(i,s).$$

We look for the least $e \leq s$ such that R_e is not yet satisfied but it can be satisfied at this stage. In other words, such that $\{e\}^{\sigma_s}_s(e)\uparrow$ and $\{e\}^\sigma_s(e)\downarrow$ for some σ that preserves σ_s up to $R(e,s)$, i.e. such that $\sigma \supseteq \sigma_s[R(e,s)]$ with $|\sigma| \leq s$.

If there is such an e pick the smallest, let σ be the smallest string as above, and let $\sigma_{s+1} = \sigma$ and

$$r(i,s+1) = \begin{cases} r(i,s) & \text{if } i < e \\ |\sigma| & \text{if } i = e \\ 0 & \text{if } i > e. \end{cases}$$

Otherwise, let $\sigma_{s+1} = \sigma_s$ and $r(i,s+1) = r(i,s)$, for every i. Let

$$A(x) = \lim_{s\to\infty} \sigma_s(x).$$

By the usual finite injury argument, $\lim_{s\to\infty} \sigma_s(x)$ exists for every x and so does $\lim_{s\to\infty} r(e,s)$. Since the construction is recursive, by the Limit Lemma we obtain $A \leq_T \mathcal{K}$.

It remains to prove by induction on e that R_e has been satisfied. Choose s large enough so that the requirements R_i with $i < e$ have been permanently satisfied before stage s, if ever, and such that $R(e,s)$ has reached its final value. If there is $\sigma \supseteq \sigma_s[R(e,s)]$ such that $\{e\}^\sigma(e)\downarrow$, then the construction will pick one up at some stage and will not change it any more, i.e. $\sigma \subseteq A$. Otherwise

$$(\forall \sigma \supseteq \sigma_s[R(e,s)])(\{e\}^\sigma(e)\uparrow),$$

and $\sigma_s[R(e,s)] \subseteq A$ by the hypothesis on s. Thus R_e is satisfied. $\square$

Permitting below r.e. degrees ⋆

We have already considered the permitting method in III.3.16, and used it in the construction of r.e. sets in various ways. We now formulate a slightly more general version that applies not only to r.e. sets, but to any set built by full approximation.

Proposition XI.2.8 Permitting method (Dekker [1954], Muchnik [1956], Friedberg [1957], Yates [1965]) *Let C be an infinite r.e. set and let f be a recursive one-one function with range C. If A is built by full approximation, i.e.*

$$A(x) = \lim_{s\to\infty} \sigma_s(x)$$

with $\sigma_s(x)$ uniformly recursive in s and x, and for every s

$$\sigma_{s+1} \supseteq \sigma_s[f(s)],$$

then $A \leq_T C$.

Proof. To see if $x \in A$, find a nondeficiency stages s of f such that $f(s) > x$ recursively in C. Then $A(x) = \sigma_s(x)$, since for $t \geq s$ we have $f(t) \geq f(s)$. Thus $\sigma_s[f(s)]$ cannot change any more. $\square$

Exercise XI.2.9 Permitting below degrees below $\mathbf{0}'$. *Let A and C be built by full approximation, with*

$$A(x) = \lim_{s\to\infty} \sigma_s(x) \quad \textit{and} \quad C(x) = \lim_{s\to\infty} \tau_s(x).$$

If $\sigma_{s+1} \supseteq \sigma_{p(s)}[m(s)]$ for every s, where

$$\begin{aligned} m(s) &= \textit{greatest } x \textit{ such that } \tau_t[x] = \tau_s[x], \textit{ for some } t < s \\ p(s) &= \textit{least } t < s \textit{ such that } \tau_t[m(s)] = \tau_s[m(s)], \end{aligned}$$

then $A \leq_T C$. (Cooper [1973]) (Hint: if t is a nondeficiency stage in the sense that, for some x, $t = \mu s.\,(\tau_s[x] = C[x])$, then $\sigma_t[m(t)] \subseteq \sigma_s$ for every sufficiently large s.)

For a discussion and applications of this type of permitting, see Posner [1981].

By applying permitting to the full approximation construction of 1-generic sets, we obtain the following result.

Proposition XI.2.10 *Every nonzero r.e. degree bounds a 1-generic degree.*

Proof. Let C be r.e. and nonrecursive, and the range of a one-one recursive function f. We keep the same notation as in XI.2.7, and add permitting. The only change in the construction is that at stage $s+1$, we look for the least $e \leq s$ such that $\{e\}_s^{\sigma_s}(e)\uparrow$ and there is σ such that

$$|\sigma| \leq s \ \wedge \ \sigma \supseteq \sigma_s[f(s)] \ \wedge \ \sigma \supseteq \sigma_s[R(e,s)] \ \wedge \ \{e\}_s^{\sigma}(e)\downarrow .$$

Since permitting only makes it harder for σ_s to change, we still have that $\lim_{s\to\infty} \sigma_s(x)$ exists and A is well-defined. Moreover, $A \leq_T C$ because we used permitting.

It remains to prove by induction on e that R_e is still satisfied. Choose s_0 large enough so that the requirements R_i with $i < e$ have been permanently satisfied before stage s_0, if ever, and such that $R(e,s)$ has reached its final value. In particular, $\sigma_s[R(e,s)] \subseteq A$ for each $s \geq s_0$. Suppose that R_e is not satisfied, i.e.

$$\{e\}^A(e)\uparrow \wedge (\forall \sigma \subseteq A)(\exists \tau \supseteq \sigma)(\{e\}^{\tau}(e)\downarrow).$$

We show that C is recursive, contradicting the hypothesis.

Given x, find $s \geq s_0$ such that for some τ we have

$$|\tau| \leq s \ \wedge \ \tau \supseteq \sigma_s[x] \ \wedge \ \tau \supseteq \sigma_s[R(e,s)] \ \wedge \ \{e\}_s^{\tau}(e)\downarrow .$$

Such an s exists by the hypothesis that R_e is not satisfied, because $\sigma_s[R(e,s)]$ is contained in A and, for a sufficiently large s, so is $\sigma_s[x]$. Note that

$$t \geq s \ \Rightarrow \ \sigma_t[x] = \sigma_s[x],$$

by induction on $t \geq s$. Indeed, by construction $\sigma_{t+1} \supseteq \sigma_t[f(t)]$. Moreover, $f(t) \geq x$ otherwise, by the induction hypothesis that $\sigma_t[x] = \sigma_s[x]$,

$$\tau \supseteq \sigma_s[x] \supseteq \sigma_t[f(t)],$$

so τ would be chosen at stage $t+1$ and R_e would be permanently satisfied.

Since the construction is recursive, s can be found recursively. Since $f(t) \geq x$ for any $t \geq s$, as just proved, we have

$$x \in C \ \Leftrightarrow \ x \in C_s.$$

Thus C is recursive, which is a contradiction. □

The theorem just proved has a number of interesting consequences.

Corollary XI.2.11 (Sacks [1963], Yates [1970a]) *If $\boldsymbol{a}$ is r.e. nonrecursive, then $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ has the following properties:*

1. *it embeds every countable partial ordering*

2. *it has minimal pairs*

3. *it is not a lattice*

4. *it has non-r.e. elements.*

Proof. The result follows from the properties of 1-generic degrees and the existence of such degrees below $\boldsymbol{a}$. $\square$

On the one hand, Sacks [1963] and Ambos-Spies [1984a] have refined parts 1 and 3 by showing that they actually hold in the r.e. degrees below $\boldsymbol{a}$. On the other hand, Lachlan [1979] has shown that part 2 does not always hold in the r.e. degrees below $\boldsymbol{a}$.

Exercises XI.2.12 a) *There is a nontrivial segment of degrees below* $\mathbf{0}'$ *not containing r.e. degrees.* (Hint: consider a 1-generic degree.)

b) *There is no segment consisting only of r.e. degrees.* (Hint: relativize XI.2.10.)

Corollary XI.2.13 (Sacks [1963a]) *Let* $\boldsymbol{a} < \mathbf{0}'$. *Then:*

1. *every countable partial ordering is embeddable in* $\boldsymbol{\mathcal{D}}([\boldsymbol{a},\mathbf{0}'])$*, and in particular* $\mathbf{0}'$ *is not a minimal cover*

2. *there are incomparable degrees* $\boldsymbol{b}$ *and* $\boldsymbol{c}$ *below* $\mathbf{0}'$ *such that* $\boldsymbol{b} \cap \boldsymbol{c} = \boldsymbol{a}$*, and thus every incomplete degree below* $\mathbf{0}'$ *is branching in* $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$*.*

Proof. By the properties of 1-generic degrees, and the relativization above $\boldsymbol{a}$ of XI.2.10 (which is possible, because $\mathbf{0}'$ is r.e. in $\boldsymbol{a}$). $\square$

Permitting below $\overline{\mathbf{GL}}_2$ degrees $\star$

Pushing 1-generic degrees below r.e. degrees makes us wonder how far we can improve on this result. On the one hand, a minimal degree does not bound a 1-generic degree. On the other hand, we already know that there are minimal degrees in $\mathbf{GL}_2$ and that not every minimal degree is in $\mathbf{GL}_1$ (V.5.14). Thus the best we can hope for is to find 1-generic degrees below any degree in $\overline{\mathbf{GL}}_2$. This is what we are now going to prove, using the characterization XI.1.3. As a corollary, it will follow that every minimal degree is in $\mathbf{GL}_2$.

Theorem XI.2.14 (Cooper [1973], Jockusch and Posner [1978]) *Every degree in* $\overline{\mathbf{GL}}_2$ *bounds a 1-generic degree.*

Proof. Let $\boldsymbol{c}$ be a degree in $\overline{\mathbf{GL}}_2$ and $C \in \boldsymbol{c}$. The idea of the proof is to build an increasing sequence of strings, at each stage making a step towards the permanent satisfaction of some requirement

$$R_e \quad : \quad \{e\}^A(e)\downarrow \vee (\exists \sigma \subseteq A)(\forall \tau \supseteq \sigma)(\{e\}^\tau(e)\uparrow),$$

as in a $\mathbf{0}'$-oracle construction. However, since we are not allowed to ask questions which are too complicated if we want our set to be recursive in C, we will bound our search by a function recursive in C. For any given requirement, we will then wait for a chance to satisfy it and when this appears we will grab it, as in a full approximation construction.

Suppose we are at stage $s+1$, with σ_s already defined. For each $e \leq s$, we can see recursively in $\mathbf{0}'$ if there is $\sigma \supseteq \sigma_s$ such that $\{e\}^{\sigma}(e)\downarrow$. Let

$$\begin{aligned} \sigma_e &= \begin{cases} \mu\sigma.\,(\sigma \supseteq \sigma_s \;\wedge\; \{e\}^{\sigma}(e)\downarrow) & \text{if } \sigma \text{ exists} \\ \emptyset & \text{otherwise} \end{cases} \\ h(s) &= \max\{|\sigma_e| : e \leq s\}. \end{aligned}$$

Thus at stage $s+1$ we can satisfy some unsatisfied R_e with $e \leq s$ exactly when we can find a string $\sigma \supseteq \sigma_s$ such that $|\sigma| \leq h(s)$ and $\{e\}^{\sigma}(e)\downarrow$.

Of course we do not know yet what σ_s is going to be, but suppose for a moment that $h(s)$ is recursive in $\mathbf{0}'$. Then, since $\boldsymbol{c} \notin \mathbf{GL}_2$, by XI.1.3 we also know that h cannot dominate every function recursive in $\boldsymbol{c}$. Then let f be a function recursive in C which is not dominated by h. For infinitely many s we will have $f(s) \geq h(s)$ and hence we could use f to bound our search, knowing that we will have infinitely many chances to satisfy our requirements.

To make h recursive in $\mathbf{0}'$ independently of the construction, we decide to have $|\sigma_s|$ recursive in $\mathbf{0}'$ as a function of s, e.g. $|\sigma_s| = s$. Then, even without knowing what σ_s is going to be, we can analyze every possibility and let

$$\begin{aligned} \sigma_e^{\tau} &= \begin{cases} \mu\sigma.\,(\sigma \supseteq \tau \;\wedge\; \{e\}^{\sigma}(e)\downarrow) & \text{if } \sigma \text{ exists} \\ \emptyset & \text{otherwise} \end{cases} \\ h(s) &= \max\{|\sigma_e^{\tau}| : e \leq s \;\wedge\; |\tau| = s\}. \end{aligned}$$

Now h is recursive in $\mathbf{0}'$ and we can fix f as above. With no loss of generality we may suppose that f is nondecreasing. The only thing to do when we would like to put σ above σ_s is to do this in many stages, instead of in just one, by extending σ_s one step at a time.

The construction is as follows. We start with $\sigma_0 = \emptyset$. At stage $s+1$, given σ_s of length s, look for the least $e \leq s$ such that $\{e\}^{\sigma_s}(e)\uparrow$ and $\{e\}^{\sigma}(e)\downarrow$ for some $\sigma \supseteq \sigma_s$ with $|\sigma| \leq f(s)$.

If there is such an e, choose the smallest one, let σ be the smallest string as above, and let σ_{s+1} be the unique string of length $s+1$ extending σ_s and contained in σ, i.e.

$$\sigma_{s+1} = \sigma_s * \langle \sigma(s) \rangle.$$

Otherwise, let $\sigma_{s+1} = \sigma_s * 0$. Thus in both cases $|\sigma_{s+1}| = s+1$.

Since f is recursive in C, $A \leq_T C$. It remains to prove by induction on e that R_e is satisfied. Choose s_0 large enough so that the requirements R_i with

$i < e$ have been permanently satisfied before stage s_0, if ever. Suppose there is no s such that $(\forall\sigma \supseteq \sigma_s)(\{e\}^\sigma(e)\uparrow)$. Since there are infinitely many s such that $f(s) \geq h(s)$, there is one such that $s \geq s_0, e$. Since there is $\sigma \supseteq \sigma_s$ such that $\{e\}^\sigma(e)\downarrow$, there is one with $|\sigma| \leq h(s) \leq f(s)$ by definition of h. Moreover, since f is nondecreasing, $|\sigma| \leq f(t)$ for any $t \geq s$. Thus either $\{e\}^{\sigma_s}(e)\downarrow$ or, at stage $s+1$, action is taken to make $\sigma \subseteq A$, since R_e has highest priority after stage s_0. □

Note that the proof shows that if f is any function not dominated by h, then there is a 1-generic set recursive in f. Since this could not happen if f had minimal degree, we have actually proved that *there is a function recursive in* $\mathbf{0}'$ *which dominates every function of minimal degree.*

Corollary XI.2.15 *If* $\mathcal{D}(\leq \boldsymbol{a})$ *is finite or a lattice, then* $\boldsymbol{a}$ *is in* $\mathbf{GL}_2$*.*

Proof. By XI.2.3 and XI.2.5. □

Exercises XI.2.16 a) *Every nonzero degree bounds a nonzero degree in* $\mathbf{GL}_2$. (Hint: 1-generic degrees are in $\mathbf{GL}_1$.)

b) *Not every nonzero degree bounds a nonzero degree in* $\mathbf{GL}_1$. (Hint: by V.5.14 there is a minimal degree that is not in $\mathbf{GL}_1$.)

Permitting below 1-generic degrees below $\mathbf{0}'$ ⋆

The goal of this subsection is to show the existence of nonzero degrees below $\mathbf{0}'$ without minimal predecessors by proving that 1-generic degrees below $\mathbf{0}'$ have this property. The idea is to consider any 1-generic degree $\boldsymbol{a}$, and show that every nonzero degree $\boldsymbol{b}$ below $\boldsymbol{a}$ bounds a 1-generic degree and thus it is not minimal.

We will use the fact that (sets recursive in) 1-generic sets below $\mathbf{0}'$ have recursive approximations that very often happen to be correct.

Proposition XI.2.17 (Shore) *If C is 1-generic and $B \leq_T C \leq_T \mathcal{K}$, then B has a recursive approximation $\{\tau_s\}_{s\in\omega}$ (called a* **correct approximation***) such that any infinite r.e. set of stages S has a member corresponding to a correct approximation of B, i.e.*

$$S \text{ r.e. and infinite} \Rightarrow (\exists s \in S)(\tau_s \subseteq B).$$

Proof. Since $C \leq_T \mathcal{K}$, there is a recursive approximation $\{\sigma_s\}_{s\in\omega}$ to it. Since $B \leq_T C$, there is i such that $B \simeq \{i\}^C$. Then $\tau_s \simeq \{i\}_s^{\sigma_s}$ (considered as a finite string) provides a recursive approximation to B.

Given S r.e. and infinite, we want $s \in S$ such that $\tau_s \subseteq B$. It is enough to find $s \in S$ such that $\sigma_s \subseteq C$, since then

$$\tau_s \simeq \{i\}_s^{\sigma_s} \subseteq \{i\}_s^C \subseteq \{i\}^C \simeq B.$$

Since S is r.e., there is an e such that

$$\{e\}^X(x) \simeq 1 \;\Leftrightarrow\; (\exists s)(s \in S \wedge \sigma_s \subseteq X),$$

i.e. $\{e\}^X$ converges exactly when there is an $s \in S$ such that σ_s is a correct approximation to X. If $\{e\}^C(x)\uparrow$ then, by genericity of C,

$$(\exists \sigma \subseteq C)(\forall \tau \supseteq \sigma)(\{e\}^\tau(x)\uparrow).$$

In particular, there is no $s \in S$ such that σ_s is a correct approximation to σ. But this is a contradiction, since $\lim_{s\to\infty} \sigma_s = C$ and S is infinite. Then $\{e\}^C(x)\downarrow$. $\square$

Theorem XI.2.18 (Chong and Jockusch [1984]) *Every nonzero degree* $\boldsymbol{b}$ *below a 1-generic degree* $\boldsymbol{c} \leq \boldsymbol{0}'$ *bounds a 1-generic degree. In other words, whenever* $\boldsymbol{c}$ *is 1-generic and* $\boldsymbol{0} < \boldsymbol{b} \leq \boldsymbol{c} \leq \boldsymbol{0}'$*, there is a 1-generic degree* $\boldsymbol{a} \leq \boldsymbol{b}$.

Proof. By the previous proposition, it is enough to prove that if B is nonrecursive and it has a correct approximation $\{\tau_s\}_{s\in\omega}$, there is a 1-generic set $A \leq_T B$.

The requirements to make A 1-generic are the usual ones:

$$R_e \quad : \quad \{e\}^A(e)\downarrow \vee (\exists \sigma \subseteq A)(\forall \tau \supseteq \sigma)(\{e\}^\tau(e)\uparrow).$$

To ensure $A \leq_T B$, we make the approximations to A follow those to B:

$$R_{-1} \quad : \quad \tau_t \subseteq \tau_s \;\Rightarrow\; \sigma_t \subseteq \sigma_s, \text{ for every } t \text{ and } s.$$

Then $A = \bigcup_{\tau_t \subseteq B} \sigma_t$ because if $\tau_t \subseteq B$, then, for all sufficiently large stages s, $\tau_t \subseteq \tau_s$, hence $\sigma_t \subseteq \sigma_s$, i.e. $\sigma_t \subseteq A$.

We try now to adapt the usual full approximation construction of XI.2.7 to also accommodate R_{-1}. At stage $s+1$, we look for the least $e \leq s$ such that $\{e\}_s^{\sigma_s}(e)\uparrow$, and for some σ

$$|\sigma| \leq s \;\wedge\; \sigma \supseteq \sigma_s[R(e,s)] \;\wedge\; \{e\}_s^{\sigma_s}(e)\downarrow.$$

When e and σ exist, we would like to let $\sigma_{s+1} = \sigma$. But if for some $t \leq s$ we have $\tau_t \subseteq \tau_{s+1}$, then we must also have $\sigma_t \subseteq \sigma_{s+1}$. Thus σ must satisfy the additional condition

$$\sigma \supseteq \sigma_t, \text{ for all } t \text{ such that } \tau_t \subseteq \tau_{s+1}.$$

There is no problem if there is more than one such t, since the respective τ_t's must be compatible (being all contained in τ_{s+1}), and hence so are the respective σ_t's, by induction on the construction.

But on top of satisfying R_{-1} for the new stage $s+1$, we obviously do not want to ruin what we have already done for its satisfaction at previous stages. This can easily be ensured by the additional request that

$$\sigma \text{ extends a maximal segment } \sigma_t, \text{ for some } t \leq s.$$

The construction is thus best viewed not in terms of strings alone, but of partial trees. At any stage, we have a finite number of finite maximal branches and we can move from a branch to another for the sake of R_{-1} and of some R_e, but only extending the current branches. Of course, for each n there will be a stage s_n such that $\sigma_s \supseteq \sigma_{s_n}[n]$ for all $s \geq s_n$, i.e. the path of the tree corresponding to A reaches a limit, because the τ_s's reach a limit, and the conditions R_e are satisfied according to their priority ordering.

The construction should now be as follows. At stage $s+1$, look for the least $e \leq s$ such that $\{e\}_s^{\sigma_s}(e)\uparrow$, and for some σ

$$|\sigma| \leq s \ \wedge \ \sigma \supseteq \sigma_s[R(e,s)] \ \wedge \ \{e\}_s^{\sigma}(e)\downarrow$$
$$\sigma \supseteq \sigma_t, \text{ for all } t \text{ such that } \tau_t \subseteq \tau_{s+1}$$
$$\sigma \text{ extends a maximal segment } \sigma_t, \text{ for some } t \leq s.$$

If e exists, let σ be the smallest string as above, and $\sigma_{s+1} = \sigma$. Otherwise, there is no R_e to satisfy at stage $s+1$, but we may not simply let $\sigma_{s+1} = \sigma_s$ as previously, because in any case we have to satisfy R_{-1}. Thus we let σ_{s+1} be any maximal string above σ_t, for all $t \leq s$ such that $\tau_t \subseteq \tau_{s+1}$.

But there is one more problem. The priority ordering by itself no longer guarantees that the requirements R_e are satisfied, the reason being that when

$$(\forall \tau \subseteq A)(\exists \sigma \supseteq \tau)(\{e\}^{\sigma}(e)\downarrow),$$

we might never be able to pick up one such σ, because of the last requirement on σ at stage $s+1$ (the other requirements are satisfied for sufficiently large stages, when the lower priority requirements have been satisfied and τ_s has settled down sufficiently).

This can easily be repaired, by requesting that whenever we try to satisfy R_e we do it at stages $s+1$ such that the upward closure of the maximal strings at that stage is the whole space, so that the last condition is vacuously satisfied for sufficiently long strings. That is, we require that

$$(\forall \sigma)(\exists t)[\sigma_t \text{ is maximal at stage } s+1 \ \wedge \ (\sigma \subseteq \sigma_t \vee \sigma \supset \sigma_t)].$$

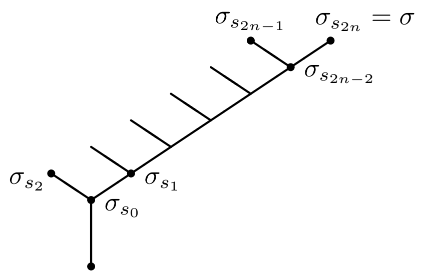
This can be achieved by the following strategy. Suppose the property is satisfied at stage $s+1$ and we would like to put up a string σ above a certain maximal σ_t. Instead of doing it all in one blow as usual, we proceed inductively one step at a time. Suppose that, at stage s_{2i}, we have $\sigma_{s_{2i-2}} \subseteq \sigma$ and we are permitted (by $\tau_{s_{2i}}$) to approach σ by one more step, by letting

$$\sigma_{s_{2i}} = \sigma_{s_{2i-2}} * \langle \sigma(|\sigma_{s_{2i-1}}|) \rangle,$$

with $\sigma_{s_{2i-1}}$ maximal at stage s_{2i}. By so doing, we choose one path in the full binary tree. If we did not do anything else, strings going in the other direction would be unusable in the following (since we only extend maximal strings at any stage). We thus wait for a stage $s_{2i+1} > s_{2i}$ until we are permitted (by $\tau_{s_{2i+1}}$) to define $\sigma_{s_{2i+1}}$ as the least string going in the other direction, i.e.

$$\sigma_{s_{2i+1}} = \sigma_{s_{2i-1}} * \langle 1 - \sigma(|\sigma_{s_{2i-1}}|) \rangle.$$

Then, at a stage $s_{2i+2} > s_{2i+1}$ when we are permitted by $\tau_{s_{2i+2}}$, we will resume our approach towards σ above $\sigma_{s_{2i}}$.



Since, at a given stage $s+1$ and for a given σ, a sequence of stages as above exists only under certain conditions (see below), we cannot simply look ahead and search for it, otherwise the construction might stall. The actual construction at stage s' will thus look back and see if for some previous stage s, condition R_e and string σ, there were stages

$$s < s_0 < s_1 < \cdots < s_{2n} < s_{2n+1} < s'$$

as above, with $\sigma_{s'} = \sigma_{s_{2n}} = \sigma$ (i.e. such that we could have successfully completed our approach toward σ, if we had pursued it). If so, we then appropriately define all strings σ_t for $s < t \leq s'$.

R_{-1} is satisfied by construction. It remains to prove by induction on e that R_e is satisfied. This rests on the facts that the approximation τ_s is correct and that B is not recursive. Suppose

$$(\forall \tau \subseteq A)(\exists \sigma \supseteq \tau)(\{e\}^{\sigma}(e)\downarrow).$$

We show that there are infinitely many s' such that $\{e\}^{\sigma_{s'}}(e)\downarrow$. By correctness of the approximation, there is one such that $\tau_{s'} \subseteq B$, hence $\sigma_{s'} \subseteq A$ and $\{e\}^A(e)\downarrow$, i.e. R_e is satisfied.

Choose s large enough so that the requirements R_i with $i < e$ have been permanently satisfied before stage s, if ever, $R(e,s)$ has reached its final value, and $\tau_s \subseteq B$. Such a stage exists by the induction hypothesis on R_i and the fact that the approximation to B is correct. By R_{-1} the construction of A will proceed from now on above σ_s, in particular $\sigma_s \subseteq A$. Thus there is $\sigma \supseteq \sigma_s$ such that $\{e\}^\sigma(e)\downarrow$. We show that there is a sequence of stages s_i as above. Since R_e is of highest priority from stage s on, and all the other conditions on σ being satisfied, σ will be put up as a (temporary) $\sigma_{s'}$ for some s', as required.

Given s as stated, wait for s_0 such that $\tau_{s_0} \subseteq B$, which exists by correctness. Then $\tau_s \subseteq \tau_{s_0}$, and we can extend σ_s by one step towards σ. Then wait for s_1 such that $\tau_s \subseteq \tau_{s_1} \not\subseteq B$, which exists otherwise B would be recursive. Then we can let σ_{s_1} extend σ_s in a way incomparable to σ_{s_0}. Then wait for s_2 such that $\tau_{s_2} \subseteq B$, which exists by correctness. Then $\tau_{s_0} \subseteq \tau_{s_2}$, and we can do one more step towards σ above σ_{s_0}, and so on. In finitely many steps, we exhaust σ. $\Box$

Corollary XI.2.19 (Lerman [1983]) *There is a nonzero degree below* $\mathbf{0}'$ *with no minimal predecessor.*

Proof. Every predecessor of a 1-generic degree below $\mathbf{0}'$ bounds a 1-generic degree and thus it is not minimal. $\Box$

We have seen two proofs of the existence of nonzero degrees in $\boldsymbol{\mathcal{D}}$ with no minimal predecessor: one by initial segments (V.6.16.c), the other by finite extensions (V.3.17). The corollary above was originally obtained by Lerman [1983] by constructivizing the first proof. Chong and Jockusch [1984] obtained their result by modifying the second proof, which does not immediately adapt because it uses two-quantifier questions and thus only produces a degree below $\mathbf{0}''$.

The direct proof given above is due to Haught [1986], who also combined it with a coding method to show that $\boldsymbol{b}$ itself must be 1-generic, and thus that *the 1-generic degrees below* $\mathbf{0}'$ *together with* $\mathbf{0}$ *are downward closed, and form an initial segment of the degrees* (in general, this fails for 1-generic Turing degrees, see XII.2.2.b, but it does hold for 1-generic m-degrees, see XI.2.6.b). However, *the 1-generic degrees (below* $\mathbf{0}'$*) do not form an ideal* because $\mathbf{0}'$ is r.e. and thus not 1-generic but, by the proof of V.2.2.6, it is the join of two 1-generic degrees.

The result that no 1-generic degree below $\mathbf{0}'$ bounds a minimal degree is the best possible. Chong and Downey [1990] and Kumabe [1990] have proved that *there are 1-generic degrees below* $\mathbf{0}''$ *that do bound minimal degrees (below* $\mathbf{0}'$*).* On the other hand, *no 2-generic degree* (see XII.1.8) *bounds a minimal degree* by the proof of V.3.17 (Jockusch [1981]). See also Chong and Downey [1989] for related results.

XI.3 Structure Theory

The properties of 1-generic degrees proved in the last section provided a good deal of knowledge about $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$: it embeds every countable partial ordering (XI.2.5.a), it is not a lattice (XI.2.5.c), it is not atomic (XI.2.19), and every incomplete degree in it is branching (XI.2.13.2). In this section we look at structure properties in a more conventional and systematic way, along the lines used for $\boldsymbol{\mathcal{D}}$ in Chapter V and for $\boldsymbol{\mathcal{R}}$ in Chapter X.

The existence of 1-generic degrees below $\mathbf{0}'$ also shows that $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ contains non-r.e. degrees (XI.2.4). It is thus natural to compare the structures of degrees below $\mathbf{0}'$ and r.e. degrees, and we will find many examples of elementary differences.

The Diamond Theorem

Recall that every countable partial ordering is embeddable both in the r.e. degrees (X.1.9) and in the degrees below $\mathbf{0}'$ (V.2.9). In particular, the one-quantifier sentences are decidable in both theories, and in the same way (V.2.10). The first place where the two theories may differ is thus at the two-quantifier level. The next result shows an elementary difference at this level.

Proposition XI.3.1 Diamond Theorem for $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ (Cooper [1972a]) *There are two nontrivial degrees* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *such that:*

$$\boldsymbol{a} \cap \boldsymbol{b} = \mathbf{0} \quad \textit{and} \quad \boldsymbol{a} \cup \boldsymbol{b} = \mathbf{0}'.$$

Proof. We want A and B recursive in $\mathcal{K}$, forming a minimal pair, and such that $A \oplus B \equiv_T \mathcal{K}$. It is not possible to follow the ideas introduced in Section V.2, because the minimal pair construction V.2.16 is incompatible with the method used in V.2.26 to force the join to be $\mathcal{K}$.

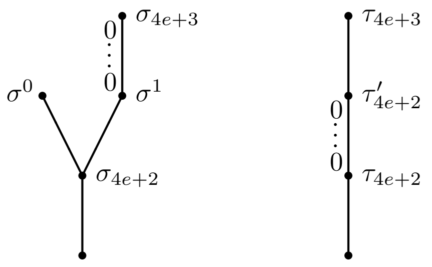
Recall that a set C is introreducible if it is recursive in each of its infinite subsets (II.6.7). It is enough to choose an introreducible set C of degree $\mathbf{0}'$ (by II.6.13) and to make $A \cap B$ an infinite subset of C to have $C \leq_T A \cap B \leq_T A \oplus B$.

The construction is now similar to that of V.2.26. We let $A = \bigcup_{s\in\omega} \sigma_s$ and $B = \bigcup_{s\in\omega} \tau_s$ and start with $\sigma_0 = \tau_0 = \emptyset$. At stage $s+1$, let σ_s and τ_s be given and of the same length.

- If $s = 4e$, we ensure that $A \not\simeq \{e\}$. Let x be the smallest element on which σ_s is undefined, and let
$$\sigma_{s+1} = \begin{cases} \sigma_s * \langle 1 - \{e\}(x) \rangle & \text{if } \{e\}(x)\downarrow \\ \sigma_s * 0 & \text{otherwise.} \end{cases}$$
In both cases, let
$$\tau_{s+1} = \tau_s * 0,$$
so that we put no new element in $A \cap B$ at this stage.

- If $s = 4e + 1$, we do the same, interchanging the roles of A and B (and hence of the σ's and τ's).

- If $s = 4e + 2$, we ensure that
$$\{e\}^A = \{e\}^B = D \quad \Rightarrow \quad D \text{ recursive.}$$
See if there are e-splitting extensions of σ_s. If not, then $\{e\}^A$ is recursive if total, so we may let
$$\sigma_{s+1} = \sigma_s \quad \text{and} \quad \tau_{s+1} = \tau_s.$$
Otherwise, choose σ^0 and σ^1 extending σ_s, and e-splitting on some x. We may suppose that σ^0 and σ^1 have the same length, otherwise we extend one of them. The usual procedure would now be to look for $\tau \supseteq \tau_s$ such that $\{e\}^\tau(x)\downarrow$. If such a τ exists, we let $\tau_{s+1} = \tau$ and choose as σ_{s+1} the σ^i $(i = 0, 1)$ such that $\{e\}^{\sigma^i}(x) \not\simeq \{e\}^\tau(x)$, which exists because σ^0 and σ^1 e-split on x. But this procedure might introduce new elements in $A \cap B$ that are not in C.
To avoid this, we first let
$$\tau_s' = \tau_s * 0 \cdots 0$$
be an extension of τ_s by 0's, of the same length as σ^i. Then see if there is $\tau \supseteq \tau_s'$ such that $\{e\}^\tau(x)\downarrow$. If not, then we can let
$$\sigma_{s+1} = \sigma^0 \quad \text{and} \quad \tau_{s+1} = \tau_s'.$$
Otherwise, choose the least such τ and let
$$\sigma_{s+1} = \sigma^i * 0 \cdots 0 \quad \text{and} \quad \tau_{s+1} = \tau,$$
with σ_{s+1} having the same length as τ_{s+1}.



- If $s = 4e+3$, then we do one step towards ensuring that $A \cap B$ is an infinite subset of C. Let x be the first element of C greater than $|\sigma_s| = |\tau_s|$, and let σ_s and τ_s be the unique strings of length $x + 1$ such that

  $$\sigma_{s+1} = \sigma_s * 0 \cdots 01 \quad \text{and} \quad \tau_{s+1} = \tau_s * 0 \cdots 01,$$

  i.e. the least extensions of σ_s and τ_s that add x and nothing else to both A and B.

By construction A and B are recursive in $\mathcal{K}$, and form a minimal pair. Moreover, $A \cap B$ is an infinite subset of C. Thus $C \leq_T A \oplus B$. $\square$

The technique used in the construction is sometimes called **intersection coding**: we succeed in coding an infinite subset of C into $A \cap B$ by placing 0's on one side whenever working on the other, except for the sake of explicitly putting an element of C into both sides.

Exercises XI.3.2 a) *The Diamond Theorem holds with $\boldsymbol{a}$ and $\boldsymbol{b}$ low.* (Epstein [1975])

b) *The Diamond Theorem holds with $\boldsymbol{a}$ and $\boldsymbol{b}$ high.* (Epstein [1975]) (Hint: treat 0 and 1 symmetrically in the construction of A and B. Also, choose an introreducible set of degree $\mathbf{0}'$ which is not contained in only finitely many columns of $\omega \times \omega$, to avoid interferences with the construction of XI.1.11.)

c) *Given $\boldsymbol{c} \geq \mathbf{0}'$, there are two nontrivial degrees $\boldsymbol{a}$ and $\boldsymbol{b}$ such that $\boldsymbol{a} \cap \boldsymbol{b} = \mathbf{0}$ and $\boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{c}$.* (Hint: by II.6.7, there is an introreducible set $C \in \boldsymbol{c}$.)

Corollary XI.3.3 (Sacks [1961a]) *$\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ and $\boldsymbol{\mathcal{R}}$ are not elementarily equivalent.*

Proof. The result follows from X.6.23 and XI.3.1. $\square$

It is interesting to note that neither the proof of the Diamond Theorem for $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ nor that of the Non-Diamond Theorem for $\boldsymbol{\mathcal{R}}$ (X.6.23) use the priority method, which is thus not required to prove that the two theories are not elementarily equivalent. Sacks' original proof used the priority method both for the existence of minimal degrees below $\mathbf{0}'$ (XI.4.5) and the nonexistence of minimal r.e. degrees (X.2.8).

From the Diamond Theorem, we also have another proof of the existence of degrees below $\mathbf{0}'$ not containing r.e. sets, since $\boldsymbol{a}$ and $\boldsymbol{b}$ cannot both be r.e. A simpler proof was given in XI.2.4, and other proofs will follow any time we find configurations of degrees below $\mathbf{0}'$ that fail in the r.e. degrees, for example minimal degrees (XI.4.5).

The Diamond Theorem shows that a simple lattice (the diamond) is embeddable in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ by preserving least and greatest elements. Fejer [1989] shows that same is true for every recursively presented lattice.

Incomparable degrees

An immediate consequence of the Diamond Theorem is the following.

Corollary XI.3.4 (Shoenfield [1959]) *Given any nontrivial degree $\boldsymbol{b} \leq \mathbf{0}'$, there is a degree $\boldsymbol{a} \leq \mathbf{0}'$ incomparable with it.*

Proof. Given two degrees $\boldsymbol{c}$ and $\boldsymbol{d}$ as in the Diamond Theorem, i.e. with g.l.b. $\mathbf{0}$ and l.u.b. $\mathbf{0}'$, they cannot be:

- one above and the other below $\boldsymbol{b}$, otherwise they would be comparable
- both above $\boldsymbol{b}$, otherwise $\mathbf{0} < \boldsymbol{b} \leq \boldsymbol{c} \cap \boldsymbol{d}$
- both below $\boldsymbol{b}$, otherwise $\boldsymbol{c} \cup \boldsymbol{d} \leq \boldsymbol{b} < \mathbf{0}'$.

Thus, at least one of $\boldsymbol{c}$ and $\boldsymbol{d}$ is incomparable with $\boldsymbol{b}$. $\square$

Exercises XI.3.5 a) *Given any nontrivial degree $\boldsymbol{b} \leq \mathbf{0}'$, there is a low degree $\boldsymbol{a} \leq \mathbf{0}'$ incomparable with it.* (Hint: by XI.3.2.a.)

b) *Given any nontrivial degree $\boldsymbol{b} \leq \mathbf{0}'$, there is a high degree $\boldsymbol{a} \leq \mathbf{0}'$ incomparable with it.* (Hint: by XI.3.2.b.)

Note that the proof of XI.3.4 is not constructive, because it does not tell which of the two given degrees is incomparable with $\boldsymbol{b}$. More precisely, it does not exhibit an effective procedure which, given an index for computing a nontrivial set B from $\mathcal{K}$, produces an index for computing a set A from $\mathcal{K}$, with A and B Turing incomparable.

We now introduce a method that will allow us to give a constructive proof of XI.3.4 in the sense just described. The obvious approach, used in V.2.13 for degrees not below $\mathbf{0}'$, is as follows. Given $B \leq_T \mathcal{K}$, we want to build $A \leq_T \mathcal{K}$ such that

$$\begin{array}{lll} R_{2e} & : & B \text{ not recursive } \Rightarrow B \not\simeq \{e\}^A \\ R_{2e+1} & : & \mathcal{K} \not\leq_T B \Rightarrow A \not\simeq \{e\}^B. \end{array}$$

R_{2e} can be treated recursively in $\mathcal{K}$, using e-splittings. R_{2e+1} is instead a source of trouble, since we would like to pick up a witness x such that $A(x)$ is still undefined, and define $A(x)$ to be different from $\{e\}^B(x)$ if this converges, and anything otherwise. But the question whether $\{e\}^B(x)\downarrow$ is only recursive in B' and hence, since $B \leq_T \mathcal{K}$, only recursive in $\mathcal{K}'$.

The familiar solution is to bound the search by functions recursive in $\mathcal{K}$. We consider one which is not dominated by any function of degree strictly below $\mathbf{0}'$ (see XI.1.5).

Proposition XI.3.6 (Yates [1967]) *Given* $\boldsymbol{b} \leq \mathbf{0}'$*, there is a uniform method of obtaining* $\boldsymbol{a} \leq \mathbf{0}'$ *such that:*

1. *if* $\mathbf{0} < \boldsymbol{b}$*, then* $\boldsymbol{b} \not\leq \boldsymbol{a}$
2. *if* $\boldsymbol{b} < \mathbf{0}'$*, then* $\boldsymbol{a} \not\leq \boldsymbol{b}$*.*

Proof. Given $B \leq_T \mathcal{K}$, we want to build $A \leq_T \mathcal{K}$ such that

$$\begin{array}{lll} R_{2e} & : & B \text{ not recursive } \Rightarrow B \not\simeq \{e\}^A \\ R_{2e+1} & : & \mathcal{K} \not\leq_T B \Rightarrow A \not\simeq \{e\}^B. \end{array}$$

By choosing $f \leq_T \mathcal{K}$ not dominated by any function of degree strictly less than $\mathbf{0}'$ (by XI.1.5), we can easily satisfy a given requirement R_{2e+1} as follows. If $\mathcal{K} \leq_T B$ or $\{e\}^B$ is not total, then there is nothing to do. If $\mathcal{K} \not\leq_T B$ and $\{e\}^B$ is total, then

$$h(x) = \mu s.\,(\{e\}^{B[s]}(x)\downarrow)$$

is a function recursive in B, and hence it does not dominate f. Thus there are infinitely many x such that $h(x) \leq f(x)$, i.e.

$$\{e\}^{B[f(x)]}(x)\downarrow .$$

It is thus enough to look for one such x on which A is still undefined, which can be done recursively in $f \leq_T \mathcal{K}$, and define

$$A(x) = 1 - \{e\}^B(x).$$

Unfortunately we do not know recursively in $\mathcal{K}$ whether $\{e\}^B$ is total, and x might not exist if $\{e\}^B$ is not total. Thus we cannot simply satisfy R_{2e+1} at a given stage. We will then do a bounded search.

R_{2e} can be satisfied recursively in $\mathcal{K}$ by looking for e-splittings in the usual way. But satisfying this type of requirement might interfere with the satisfaction of some R_{2e+1}, because action for the former commits elements into A or $\overline{A}$, and we might always commit in the wrong way all x's such that $\{e\}^{B[f(x)]}(x)\downarrow$, before we discover that we can use them to satisfy R_{2e+1}. A little care is thus needed in putting together the strategies for the two types of requirements.

We can proceed in two ways:

1. *first proof*
   Instead of brutally satisfying R_{2e} whenever possible, we slow down the construction as we did in XI.2.14, by requesting that $|\sigma_s| = s$. Thus at stage $s+1$, we can check whether s is a possible witness for some R_{2e+1}.

   The construction is as follows. We start with $\sigma_0 = \emptyset$. At stage $s+1$, given σ_s of length s, look for the least $i \leq s$ such that R_i is still unsatisfied and may be satisfied now, and take action to satisfy it. Precisely:

   - R_{2e} may be satisfied at stage $s+1$ if there are extensions σ^j $(j = 0, 1)$ of σ_s, e-splitting on some x. This can be answered recursively in $\mathcal{K}$. If so, to satisfy R_{2e} one chooses the least such pair and j such that $\{e\}^{\sigma^j}(x) \not\simeq B(x)$. Let
   $$\sigma_{s+1} = \sigma_s * \sigma^j(s).$$
   At later stages we will keep on pushing σ_s along σ^j, as long as R_{2e} has highest priority, until we exhaust σ^j and thus satisfy R_{2e}.
   - R_{2e+1} may be satisfied at stage $s+1$ if $\{e\}^{B[f(s)]}(s)\downarrow$. To satisfy it, let
   $$\sigma_{s+1} = \sigma_s * \langle 1 - \{e\}^B(s)\rangle.$$

   If there is no such i, let $\sigma_{s+1} = \sigma_s$.

   A simple induction on i shows that all R_i's are satisfied.

2. *second proof*
   We decide to satisfy all requirements R_{2e} before we even begin to worry about the R_{2e+1}'s. Of course we have to leave enough space to satisfy the latter and thus we build, recursively in $\mathcal{K}$, a tree T (we keep the usual notation for trees, see V.5.1) all of whose branches satisfy all requirements R_{2e}. This can be done by standard methods using e-splittings for R_{2e}.

   We now choose a branch of T satisfying the R_{2e+1}'s. Let
   $$m(s) = \max\,\{|T(\sigma)| : |\sigma| = s\}$$

be the maximum length of strings of T of level s. By XI.1.6 there is a function $f \leq_T \mathcal{K}$ which is not dominated by any function recursive in B on $S = \{m(s)\}_{s\in\omega}$.

The construction is as follows. We start with $\sigma_0 = \emptyset$. At stage $s+1$, given $\sigma_s = T(\sigma)$ with $|\sigma| = s$ (i.e. σ_s is at level s of the tree T), we want to decide which way to go. Look for the least $e \leq s$ such that R_{2e+1} has not yet been satisfied and $\{e\}^{B[f(s+1)]}(x)$ converges for every $x \leq m(s+1)$.

If there is such an e, since $T(\sigma * 0)$ and $T(\sigma * 1)$ are incompatible and both of length $\leq m(s+1)$, one of them is incompatible with $\{e\}^{B[f(s+1)]}$. Let it be σ_{s+1}. Otherwise, let e.g. σ_{s+1} be $T(\sigma * 0)$.

It remains to prove by induction on e that R_{2e+1} is satisfied. Choose s_0 large enough so that every R_{2i+1} with $i < e$ has been satisfied before stage s_0, if ever. If $\{e\}^B$ is total, there must be a stage $s \geq s_0$ such that R_{2e+1} is satisfied at stage $s+1$. Otherwise, as R_{2e+1} is of highest priority after stage s_0, the function

$$h(y) = \mu z.\, (\forall x \leq y)(\{e\}^{B[z]}(x)\downarrow)$$

would be recursive in B and would dominate f on S, which is a contradiction. $\Box$

A further way of solving the problems posed by the requirements of the last result is given in the next subsection (see the proof of XI.3.9).

Exercises XI.3.7 a) *There is a degree below* $\mathbf{0}'$ *incomparable with every nontrivial r.e. degree.* (Yates [1967]) (Hint: given a sequence of sets uniformly recursive in $\mathcal{K}$, there is a set recursive in $\mathcal{K}$ incomparable with every nontrivial element of the sequence.)

b) *Given a nontrivial degree below* $\mathbf{0}'$*, there is an r.e. degree incomparable with it.* (Sacks [1963a]) (Hint: modify the proofs of X.6.14 or X.6.1 by using the modified length of agreement as in X.1.11.a, so that the negative requirements can be dealt with not only for given nonrecursive r.e. sets, but also for nonrecursive Δ^0_2 sets.)

Exercises XI.3.8 Chains and antichains below $\mathbf{0}'$. Recall (V.2.19) that a chain is a linearly ordered set of degrees and an antichain is a set of nontrivial pairwise incomparable degrees.

a) *Every finite chain of degrees below* $\mathbf{0}'$ *is extendable and thus every maximal chain is countable.* (Hint: see XI.2.13.1.)

b) *Every finite antichain of degrees below* $\mathbf{0}'$ *is extendable and thus every maximal antichain is countable.* (Hint: given finitely many nontrivial degrees below $\mathbf{0}'$, there is another incomparable with them.)

The Capping Theorem

We already know, for example from XI.2.3, that there are minimal pairs of degrees below $\mathbf{0}'$. Actually, much more is true.

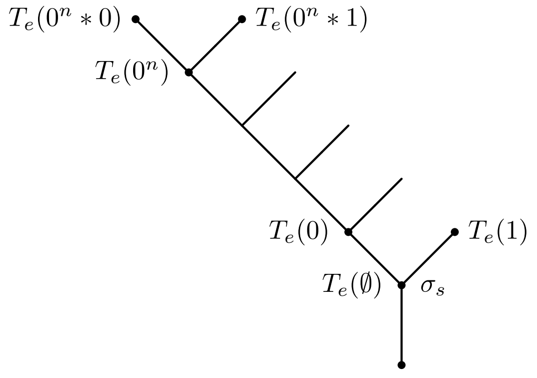
Theorem XI.3.9 Capping Theorem for $\mathcal{D}(\leq \mathbf{0}')$ (Shoenfield [1966]) *Every nontrivial degree below* $\mathbf{0}'$ *is part of a minimal pair.*

Proof. Given $B <_T \mathcal{K}$, we want to build $A \leq_T \mathcal{K}$ such that

$$\begin{array}{lcl} R_{2e} & : & A \not\simeq \{e\} \\ R_{2e+1} & : & \{e\}^A \simeq \{e\}^B \simeq C \;\Rightarrow\; C \text{ recursive.} \end{array}$$

The obvious attack on R_{2e+1}, used in V.2.16 for degrees not below $\mathbf{0}'$, is as follows. At a given stage s, given σ_s, we look for e-splitting extensions of it. If they do not exist, then, for any set A extending σ_s, $\{e\}^A$ will be recursive if total. If they do exist, we choose the one that produces a disagreement with $\{e\}^B$ on the x on which they e-split. The obvious trouble with this is that we cannot ask recursively in $\mathcal{K}$ whether $\{e\}^B(x)$ converges, because this question is only recursive in $\mathcal{K}'$ (since B is recursive in $\mathcal{K}$).

The idea is to consider a function $f \leq_T \mathcal{K}$ not dominated by any function of degree strictly less than $\mathbf{0}'$ (by XI.1.5), and hence not by any function recursive in B, and to use it to bound the search for convergent computations of $\{e\}^B$. Precisely, given σ_s we build (recursively in $\mathcal{K}$) a partial tree T_e of strings extending it and growing along its leftmost branch as in the picture.



We let $T_e(\emptyset) = \sigma_s$. Given a string $T_e(0^n)$ (with 0^n the sequence consisting of n zeros), at level n and on the leftmost branch, we ask recursively in $\mathcal{K}$ whether it has e- splitting extensions. If not, then we have found an extension of σ_s without e-splitting extensions, and we can let it be σ_{s+1}. Then $\{e\}^A$ is going to be recursive if total, and we stop building T_e. If there are e-splitting extensions of $T_e(0^n)$, we choose two and put them on the tree at the next level as $T_e(0^n * i)$ $(i = 0, 1)$, as possible candidates for the satisfaction of R_{2e+1}. We then ask (again recursively in $\mathcal{K}$) whether $\{e\}^B$ converges in less than $f(n)$ steps on the element x_e^n on which they e-split. If so, we can let σ_{s+1} be the one of $T_e(0^n * i)$ that produces a disagreement with $\{e\}^B$ on x_e^n. Since then $\{e\}^A \not\simeq \{e\}^B$, we stop building T_e.

The crucial point would be to show that if $\{e\}^B$ is total, one of the two cases must eventually happen. Suppose the tree T_e is infinite. Then x_e^n is defined for every n, and the function

$$g(n) \;=\; \mu s.\,(\{e\}_s^B(x_e^n)\downarrow)$$

is total because $\{e\}^B$ is total. Moreover, g dominates f because by hypothesis the last case of the construction never happens (otherwise T_e would be finite). If g were recursive in B we would have the desired contradiction (by the choice of f), but we only know that g is recursive in $\mathcal{K}$, because it uses the sequence $\{x_e^n\}_{n\in\omega}$ obtained from the construction (which is recursive in $\mathcal{K}$). To eliminate reference to the x_e^n's, we can define

$$\begin{aligned} h(0) &= g(0) \\ h(n+1) &= \mu s.\,(\{e\}_s^B[h(n)]\downarrow), \end{aligned}$$

i.e. in each successor point we majorize the number of steps needed to compute $\{e\}^B$ on the whole segment determined by the previous value of h. Now we do have $h \leq_T B$, and it would be enough to have $g(n) \leq h(n)$. This holds by definition for $n = 0$, and it would hold for $n + 1$ if we had $x_e^{n+1} < g(n)$, since then $x_e^{n+1} < g(n) \leq h(n)$ by the inductive hypothesis, and hence

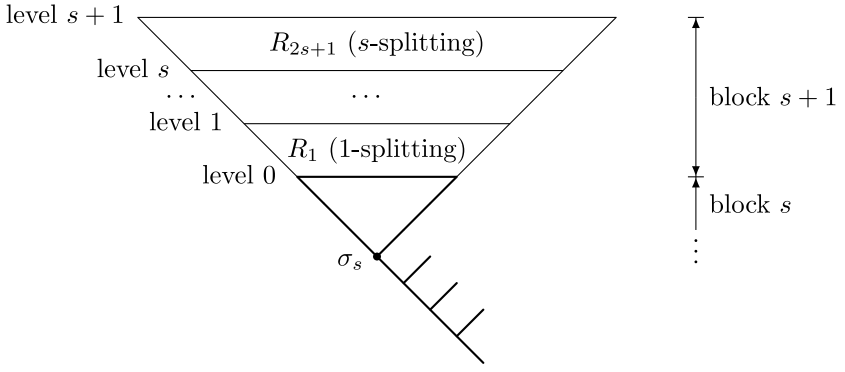
$$h(n+1) \;\geq\; g(n+1) \;=\; \mu s.\,(\{e\}_s^B(x_e^{n+1})\downarrow).$$

We can ensure $x_e^{n+1} < g(n)$ by a small change in the construction of the tree T_e. At each level n, when working on $T_e(0^n)$ and the element x_e^n on which $T_e(0^n * 0)$ and $T_e(0^n * 1)$ e-split, we have to look ahead one more step. We define the strings $T_e(0^{n+1} * 0)$ and $T_e(0^{n+1} * 1)$ above $T_e(0^{n+1})$, find the element x_e^{n+1} on which they e-split, and see whether $\{e\}^B$ converges on x_e^n in less than

$$\max\,\{f(n), x_e^{n+1}\}$$

steps. If so, we can define σ_{s+1} as above. If not, this produces $x_e^{n+1} < g(n)$ as required.

If we could tell recursively in $\mathcal{K}$ whether $\{e\}^B$ is total or not, then we could just consider the requirements R_{2e+1} for which $\{e\}^B$ is total and satisfy them at given stages, since we would know that one of the cases described above is going to eventually happen. As this is not possible, we have to work simultaneously for all requirements and satisfy them when the chance arises. Since the action devoted to any requirement is potentially infinite, we dovetail action for each of them: at stage $s+1$ we will consider the requirements R_{2e+1} with $e \leq s$ that have not yet been satisfied, and for each of them successively add e-splitting strings whenever possible. If we insist in using the simple approach described above, with a tree growing along its leftmost branch, then we run into trouble as follows. Whenever we decide to leave the leftmost branch because of a successful diagonalization step, by letting σ_{s+1} be some string $T_e(0^n * 1)$, we discharge all splittings that we might have put above $T_e(0^n * 0)$. This interferes with the preceding work, since each discharged splitting provided an element x_e^{m+1} used to bound the computation of a value of $\{e\}^A$ on an element x_e^m provided by the previous level. We thus injure all the requirements already considered and not yet satisfied, with no control over the number of possible injuries. To avoid this, it is enough to consider more general trees, instead of the simple ones used until now. If the branch $T_e(0^n * 1)$ had splittings of its own above it instead of being a dead end, and if an e-splitting on x at a given level used the maximum of $f(s)$ and of all the elements of the e-splittings at the next level as a bound of the computation of $\{e\}^B(x)$, then going above $T_e(0^n * 1)$ would cause less problems.



The construction is then as follows. We start with $\sigma_0 = \emptyset$. At stage $s+1$, we have a finite global tree (representing the work done until now and not injured for all requirements considered, and of which the various T_e are subtrees) and a string σ_s on it. The only part of the tree that matters, which is boldface in the picture and to which we restrict attention, consists of the splittings along σ_s (as before) and the subtree above σ_s, the latter being a part of auxiliary blocks built at previous stages. We now build block $s+1$, which consists of $s+1$ levels. Let level 0 be the given tree, and at each level $e \leq s$ play one step of the strategy for R_{2e+1} above any given string at level e, by either determining that there is no e-splitting above it, or adding e-splittings if they exist.

The strings at level e of this block provide the $(n+1)$-st level of the tree T_e, if T_e has been defined up to level n at the end of stage s, and its 0-th level otherwise (in other words, level n of T_e consists of the strings of level e of block $n+e+1$, due to the fact that R_e gets attention for the first time at stage $e+1$). In the first case, we now have elements on which the strings of level $n+1$ of T_e e-split, and we can use the maximum among them and $f(n)$ to bound the computations of $\{e\}^B$ on the elements on which the strings at level n of T_e e-split. We thus determine if there is R_{2e+1} $(e \leq s)$ that can be satisfied at this stage. If so, we decide to satisfy the one with highest priority (i.e. lowest index), and define σ_{s+1} accordingly. Otherwise, we can let e.g. σ_{s+1} be the leftmost string of the s-th block.

The reason we can take the last step when no requirement can be satisfied at stage $s+1$ is that block $s+1$ alone (together with f) decides whether strings of block s can be used to diagonalize, and successive work cannot change the decision. Since this was not the case, any string of block s above σ_s is good.

The satisfaction of some R_{2e+1} of highest priority at stage $s+1$ presents a final difficulty. Recall that there are two possibilities: either σ_{s+1} has been chosen because it has no e-splitting extensions (in which case this was discovered at the present stage, and thus σ_{s+1} is in block $s+1$), or σ_{s+1} is one side of an e-splitting pair (in which case it was discovered that it could be used only after the definition of block $s+1$, but σ_{s+1} itself is in block s). The difficulty occurs when R_{2e+1} has been satisfied by σ_{s+1} (at level e of block $s+1$) because of the first case. Consider the i-splittings along σ_{s+1} at lower levels $i < e$ of block $s+1$. If we discovered, when building the next block $s+2$, that they could be used, then on the one hand we could not do so because we are already committed to σ_{s+1}, and on the other hand they would no longer witness that some computation of $\{e\}^B$ is sufficiently long. Thus they have to be dropped and the trees relative to R_{2i+1} with $i < e$ have to be started anew. But this might happen infinitely many times, for any $e > i$, because we drop the trees relative to a given requirement when a requirement of *lower* priority is satisfied. However, the solution is at hand. It is enough to define the blocks with inverted levels, i.e. by working at stage $s+1$ with the requirements R_{2i+1} in inverted order, starting with R_{2s+1} at level 0 and proceeding to R_1. The difficulty just discussed now disappears, if we drop the trees relative to a given requirement whenever a *higher* priority requirement is satisfied, and thus trees for a given requirement can change only finitely often.

This takes care of the requirements R_{2e+1}, but we have not yet done anything for the requirements R_{2e}. Fortunately, there is no problem here. Since we only want to diagonalize, we can simply modify the previous construction by adding one more level to block $s+1$, extending each string at level $s+1$ to a string that diagonalizes against the s-th recursive function. This can be done recursively in $\mathcal{K}$ and without interfering with the rest of the construction, except for slowing it down. $\square$

A different and simpler proof of XI.3.9 follows from XI.4.6.a, since a minimal degree incomparable with $\boldsymbol{b}$ obviously forms a minimal pair with it. The advantage of the present proof (due to Slaman and Steel [1989]) is that it is compatible with the proof of XI.3.10, and it thus extends to a proof of the Complementation Theorem XI.3.13.

Yates [1967] has proved the **Non-Capping Theorem for $\boldsymbol{\mathcal{R}}$** (see X.6.9.a and X.6.24), which provides another elementary difference between the two structures $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ and $\boldsymbol{\mathcal{R}}$.

The Cupping Theorem

We already know from V.2.26 that there are pairs of nontrivial degrees joining to $\mathbf{0}'$. Actually, much more is true.

Theorem XI.3.10 Cupping Theorem for $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ (Posner and Robinson [1981]) *Every nontrivial degree below* $\mathbf{0}'$ *is part of a pair joining to* $\mathbf{0}'$.

Proof. Given a nonrecursive set B we build a 1-generic set A such that $A \oplus B \equiv_T \mathcal{K}$. The construction will give $A \leq_T \mathcal{K}$, and by 1-genericity we will have $\mathcal{K} \not\leq_T A$ (because then $A' \equiv_T \mathcal{K}$). To make $\mathcal{K} \leq_T A \oplus B$ we will code $\mathcal{K}$ into A and make sure that the construction is recursive in $A \oplus B$, so that we can decode $\mathcal{K}$ recursively in $A \oplus B$. To make A 1-generic and code $\mathcal{K}$ into it is no problem (see V.2.24), so we only have to worry about making the construction recursive in $A \oplus B$. For this purpose, we fix a recursive sequence $\{\tau_n\}_{n\in\omega}$ of mutually incompatible strings, e.g.

$$\tau_n = \underbrace{0\cdots 0}_{n \text{ times}} 1.$$

The construction is as follows. We start with $\sigma_0 = \emptyset$. At stage $e+1$, given σ_e, we look for the smallest n such that

$$\begin{array}{l} n \notin B \ \wedge\ (\exists \sigma \supseteq \sigma_e * \tau_n)(\{e\}^\sigma(e)\downarrow) \text{ or} \\ n \in B \ \wedge\ (\forall \sigma \supseteq \sigma_e * \tau_n)(\{e\}^\sigma(e)\uparrow). \end{array}$$

If such an n exists, take the smallest one. If the first case happens, take the smallest σ as above, and let

$$\sigma_{e+1} = \begin{cases} \sigma * \langle \mathcal{K}(e) \rangle & \text{if the first case happens} \\ \sigma_e * \tau_n * \langle \mathcal{K}(e)\rangle & \text{otherwise.} \end{cases}$$

To prove that such an n always does exist, note that

$$S \;=\; \{n : (\exists \sigma \supseteq \sigma_e * \tau_n)(\{e\}^{\sigma}(e)\downarrow)\}$$

is an r.e. set. If no such n exists, $B = S$, so B is r.e. However, we may assume without loss of generality that B is not r.e.: we are only interested in its degree, and $B \oplus \overline{B}$ has the same degree as B, and is not r.e. (otherwise B and $\overline{B}$ would both be r.e. and B would be recursive).

Thus an n as above exists and it can be found recursively in $\mathcal{K}$, because $B \leq_T \mathcal{K}$, and only one-quantifier questions are asked. In particular, $A \leq_T \mathcal{K}$ and thus $A \oplus B \leq_T \mathcal{K}$.

To show that $\mathcal{K} \leq_T A \oplus B$, note that

$$e \in \mathcal{K} \;\Leftrightarrow\; \sigma_{e+1}(|\sigma_{e+1}|) = 1.$$

Thus it is enough to show that the construction is recursive in $A \oplus B$. At step $e+1$, look for n such that $\sigma_e * \tau_n \subseteq A$. This requires only A, and n is unique because the τ_n's are incompatible. Then see if $n \in B$. If so, σ_{e+1} is a one-element extension of $\sigma_e * \tau_n$, which can be obtained knowing A. Otherwise, look recursively for the smallest $\sigma \supseteq \sigma_e * \tau_n$ such that $\{e\}^{\sigma}(e)\downarrow$. Then σ_{e+1} is a one-element extension of σ. $\Box$

Yates and Cooper have proved the **Non-Cupping Theorem for $\boldsymbol{\mathcal{R}}$** (see X.6.17), which provides another elementary difference between the two structures $\boldsymbol{\mathcal{D}(\leq 0')}$ and $\boldsymbol{\mathcal{R}}$.

For future use, notice that if we replace $\mathcal{K}$ by any set C, then the same proof produces a combination of the Jump Inversion Theorem and a generalization of the Cupping Theorem for every degree above $\mathbf{0}'$: *if $\mathcal{K} \leq_T C$ and $B \leq_T C$ is not recursive, then there is an A such that*

$$A' \;\equiv_T\; A \oplus B \;\equiv_T\; C.$$

The second part is proved as above. For the first part, one only needs to note that $A' \leq_T A \oplus B$ because

$$e \in A' \;\Leftrightarrow\; \{e\}^{\sigma_{e+1}}(e)\downarrow,$$

and that $C \leq_T A'$ because C is recursive in A and in the construction, and the latter is recursive in A' by the same argument as in the last part of the proof.

Exercises XI.3.11 a) *Every degree not above $\mathbf{0}'$ has a successor in $\mathbf{GL}_1$.* (Jockusch) (Hint: given $\boldsymbol{b} \not\geq \mathbf{0}'$ not in $\mathbf{GL}_1$, then $\boldsymbol{b} < \boldsymbol{b} \cup \mathbf{0}' < \boldsymbol{b}'$. Relativize the Cupping Property to $\boldsymbol{b}$, to obtain $\boldsymbol{a} \geq \boldsymbol{b}$ such that $\boldsymbol{a}' = \boldsymbol{b}'$ and $\boldsymbol{a} \cup \boldsymbol{b} \cup \mathbf{0}' = \boldsymbol{b}' = \boldsymbol{a}'$. Thus $\boldsymbol{a}$ is in $\mathbf{GL}_1$.)

b) *$\mathbf{0}'$ is the least degree without successors in $\mathbf{GL}_1$, as well as the least degree with all successors in $\mathbf{GH}_1$.*

The Complementation Theorem

By the Capping and Cupping Theorems, for every nontrivial degree $\boldsymbol{b}$ below $\mathbf{0}'$ there are two nontrivial degrees $\boldsymbol{a}$ and $\boldsymbol{c}$ below $\mathbf{0}'$ such that

$$\boldsymbol{a} \cap \boldsymbol{b} = \mathbf{0} \text{ and } \boldsymbol{c} \cup \boldsymbol{b} = \mathbf{0}'.$$

We now combine the two proofs and show that $\boldsymbol{a}$ and $\boldsymbol{c}$ can actually be chosen as the same degree. This shows that $\boldsymbol{b}$ has a complement in the degrees below $\mathbf{0}'$.

Exercise XI.3.12 *Every degree in* $\mathbf{L}_2$ *has a complement.* (Epstein [1975], Posner and Robinson [1981]) (Hint: by XI.1.4.d, the sets recursive in a low$_2$ degree are uniformly recursive in $\mathbf{0}'$. Then we can diagonalize directly against them during the proof of the Cupping Theorem XI.3.10, thus producing a complement. We only have to be careful in the diagonalization, by looking for an n such that either $n \notin B$ and there are e-splitting extensions of $\sigma_s * \tau_n$, or $n \in B$ and there are no e-splitting extensions of $\sigma_s * \tau_n$.)

Theorem XI.3.13 Complementation Theorem for $\mathcal{D}(\leq \mathbf{0}')$ (Posner and Robinson [1981], Posner [1981a]) *Every degree below* $\mathbf{0}'$ *has a complement.*

Proof. All of the hard work has already been done in the proofs of XI.3.9 and XI.3.10, whose notation and hypotheses we use here. We just have to note that those proofs are compatible.

First, recall from XI.3.10 that we need the construction to be recursive in $A \oplus B$. We use the same trick used there: we do not look directly for the strings we need, but rather for strings whose outcome is coded by (non)membership of certain elements in B. Precisely, we modify the construction of XI.3.9 by replacing each step in which we look for e-splitting extensions of a string τ by a step in which we look for the smallest n such that

$$\begin{array}{l} n \notin B \ \wedge \ \tau * \tau_n \text{ has } e\text{-splitting extensions, or} \\ n \in B \ \wedge \ \tau * \tau_n \text{ has no } e\text{-splitting extension.} \end{array}$$

In the first case we take two such extensions of $\tau * \tau_n$, and in the second case we consider $\tau * \tau_n$.

We then need to insert specific steps to force $A \oplus B \equiv_T \mathcal{K}$, as in XI.3.10. This is done exactly as we did for diagonalization at the end of XI.3.9, by inserting at each stage $s+1$ in the construction of XI.3.9 a step similar to that of stage $s+1$ in XI.3.10. Precisely, we add to the construction of XI.3.9 modified as above one more level to block $s+1$, extending each string τ at level $s+1$ to a string

$$\sigma * \langle \mathcal{K}(s) \rangle,$$

obtained as follows. We look for the smallest n such that

$$\begin{array}{l} n \notin B \ \wedge \ (\exists \sigma \supseteq \tau * \tau_n)(\{e\}^\sigma(e)\downarrow) \text{ or} \\ n \in B \ \wedge \ (\forall \sigma \supseteq \tau * \tau_n)(\{e\}^\sigma(e)\uparrow). \end{array}$$

Then we let σ be the smallest string as above if the first case happens, and $\tau * \tau_n$ otherwise.

The modifications and additions described above can be done recursively in $\mathcal{K}$ and without interfering with the rest of the construction, except for slowing it down. The proof that $A \oplus B \equiv_T \mathcal{K}$ is then as in XI.3.10. $\square$

The Complementation Theorem for $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ was obtained after a number of partial results, proving complementation for the subclasses $\boldsymbol{\mathcal{R}}$ (Epstein [1975]), $\mathbf{L}_2$ (Posner and Robinson [1981]), $\mathbf{H}_1$ (Posner [1977]), and $\overline{\mathbf{L}}_2$ (Posner [1981a]). In particular, Posner and Robinson [1981] and Posner [1981a] together provided a nonuniform proof of XI.3.13. The uniform proof given above is due to Slaman and Steel [1989].

Seetapun and Slaman [199?] have proved that *every nontrivial degree below* $\mathbf{0}'$ *has a complement of minimal degree*. Minimal complements had previously been obtained for degrees in $\boldsymbol{\mathcal{R}}$ (Epstein [1975]) and $\mathbf{H}_1$ (Posner [1977]).

Exercise XI.3.14 *The Complementation Theorem holds in* $\boldsymbol{\mathcal{D}}(\leq \mathbf{c})$, *for any* $\mathbf{c} \geq \mathbf{0}'$. (Posner [1981]) (Hint: given C such that $\mathcal{K} \leq_T C$, consider $B <_T C$. There are two cases, according to whether $\mathcal{K}$ is recursive in B or not. In the latter, the function f recursive in $\mathcal{K}$ defined as

$$f(x) = \begin{cases} \mu s. \, (x \in \mathcal{K}_s) & \text{if } x \in \mathcal{K} \\ 0 & \text{otherwise} \end{cases}$$

is not dominated by any function recursive in B, otherwise $\mathcal{K} \leq_T B$. Then the same proof of the theorem above, using f as in XI.3.9, produces a complement for B below C. If instead $\mathcal{K} \leq_T B$, then one builds a nonrecursive A such that $D \oplus B \equiv_T C$ for each nonrecursive $D \leq_T A$, so that $A \oplus B \equiv_T C$, and only recursive sets can be below both A and B. Thus A and B are complements below C. To obtain A, build a tree $T \leq_T \mathcal{K}$ such that, for every string σ at level n, for every $e \leq n$ either $T(\sigma)$ has no e-splitting extensions, or $T(\sigma * 0)$ and $T(\sigma * 1)$ e-split. Moreover, $T(\sigma)$ is different from the n-th recursive function. If A is the path of T determined by C, then, when $\{e\}^A$ is total and nonrecursive, $\{e\}^A \oplus \mathcal{K} \equiv_T C$, and hence $\{e\}^A \oplus B \equiv_T C$ because $\mathcal{K} \leq_T B \leq_T C$.)

Exact pairs and ideals

We now look at analogues of the results of Section V.4. We have already noticed in V.4.6 that there are $2^{\aleph_0}$ ideals of degrees in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$. Thus not every such ideal can have exact pairs below $\mathbf{0}'$. The next observation computes a bound on the complexity of ideals that do.

Proposition XI.3.15 (Yates [1970]) *If $\mathcal{I}$ is an ideal with an exact pair below $\mathbf{0}'$, then there is a Σ^0_4 set I such that*

$$e \in I \;\Leftrightarrow\; deg(\{e\}^{\mathcal{K}}) \in \mathcal{I}.$$

Proof. Let $A, B \leq_T \mathcal{K}$ be sets whose degrees are an exact pair for $\mathcal{I}$. Then the degree of $\{e\}^{\mathcal{K}}$ is in $\mathcal{I}$ if and only if

$$\{e\}^{\mathcal{K}} \text{ total } \wedge \; \{e\}^{\mathcal{K}} \leq_T A \; \wedge \; \{e\}^{\mathcal{K}} \leq_T B.$$

The first condition is Π^0_2 in $\mathcal{K}$, the rest is Σ^0_3 in $A \oplus B \oplus \mathcal{K}$ and hence in $\mathcal{K}$, i.e. everything is Σ^0_4. $\square$

Exercises XI.3.16 a) *A principal ideal in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ is Σ^0_4.*

b) *A principal ideal in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ that is a lattice is Σ^0_3.* (Hint: the sets recursive in A are $\Sigma^{0,A}_3$. By XI.2.15, if the degrees below the degree of A are a lattice, then A is low$_2$ and $\Sigma^{0,A}_3 = \Sigma^0_3$.)

Thus only Σ^0_4 ideals in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ can have exact pairs below $\mathbf{0}'$. Moreover, if such an exact pair exists, then the ideal must be bounded below $\mathbf{0}'$.

The proof of Spector's Theorem (V.4.3) shows that any ideal $\{B_n\}_{n\in\omega}$ of uniformly low degrees (i.e. such that $(\oplus_{n\in\omega} B_n)' \leq_T \mathcal{K}$) has an exact pair below $\mathbf{0}'$. Alternatively, every ideal of degrees bounded by a low degree $\boldsymbol{e}$ and (with set of indices) uniformly recursive in it has an exact pair below $\mathbf{0}'$. The next result is a three-quantifier improvement of (the proof of) V.4.3.

Theorem XI.3.17 (Nerode and Shore [1980a], Shore [1981a]) *If $\mathcal{I}$ is an ideal bounded by a low degree $\boldsymbol{e}$ and Σ^0_3 in it, then it has an exact pair below $\mathbf{0}'$.*

Proof. Let E be a set of degree $\boldsymbol{e}$ and let I be a $\Sigma^{0,E}_3$ set of indices such that the degree of $\{e\}^E$ is in $\mathcal{I}$ if and only if $e \in I$. Then there is a predicate Q recursive in E such that

$$e \in I \;\Leftrightarrow\; (\exists x)(\forall y)(\exists z)Q(x, y, z, e).$$

We want to build A and B such that

$$\begin{aligned} C_e &\;:\; e \in I \;\Rightarrow\; \{e\}^E \leq_T A, B \\ R_e &\;:\; \{e\}^A \simeq \{e\}^B \simeq C \;\Rightarrow\; C \leq_T \Big(\bigoplus\nolimits_{m\leq n \wedge m\in I} \{m\}^E\Big), \text{ for some } n. \end{aligned}$$

We will construct A and B by finite initial segments σ_s and τ_s, respectively. The construction will be a full approximation recursive in E, i.e. $\sigma_s(x)$ and $\tau_s(x)$ will be uniformly recursive in E and

$$A(x) = \lim_{s\to\infty} \sigma_s(x) \qquad B(x) = \lim_{s\to\infty} \tau_s(x).$$

This makes $A, B \leq_T E'$, by the Limit Lemma. By the hypothesis of lowness of E, A and B are then recursive in $\mathcal{K}$.

The coding requirement C_e will be satisfied by coding $\{e\}^E$ into a column of both A and B, with at most finitely many exceptions. First, we split C_e into infinitely many requirements, one for each possible x witnessing that $e \in I$:

$$C_{e,x} \ : \ (\forall y)(\exists z)Q(x,y,z,e) \Rightarrow \{e\}^E \leq_T A, B.$$

To satisfy $C_{e,x}$, we will code $\{e\}^E$ in the $\langle e,x\rangle$-th column of both A and B. This will require infinite action, since there are infinitely many values to code. We will then again split $C_{e,x}$ into infinitely many requirements

$$C_{e,x,n} \ : \ (\forall y \leq n)(\exists z)Q(x,y,z,e) \Rightarrow \{e\}^E(n) \text{ gets coded into } A \text{ and } B.$$

To satisfy $C_{e,x,n}$ at stage $s+1$, when $(\forall y \leq n)(\exists z \leq s)Q(x,y,z,e)$ and $\{e\}^E(n)$ converges in at most s steps (both these conditions being recursive in E), we put $\langle n, \{e\}^E(n)\rangle$ in the $\langle e,x\rangle$-th column of both A and B. If all $C_{e,x,n}$ are satisfied and $e \in I$, then $\{e\}^E \leq_T A, B$, since to compute $\{e\}^E(n)$ it is enough to look for an element $\langle n, w\rangle$ on the $\langle e,x\rangle$-th column.

The exactness requirement R_e will be satisfied as in the usual minimal pair construction, by trying to make the two sides $\{e\}^A$ and $\{e\}^B$ different. Since we work recursively in E, we cannot simply satisfy each of these requirements at a given fixed stage, because to ask whether there are e-splitting strings compatible with the work already done is only recursive in $\mathcal{K}$. We thus wait for these strings to appear, and grab them whenever we can. To be able to satisfy all requirements R_e, we will always try to satisfy the one with smallest index that can be satisfied. This introduces a priority ordering, as well as the possibility of changing a string that was chosen for the satisfaction of some requirement, by preferring a string that would satisfy a requirement with smaller index (i.e. of higher priority). To allow the columns of A and B to settle, we will not let R_e interfere with columns of index $\leq e$.

The construction is as follows. We start with $\sigma_0 = \tau_0 = \emptyset$. At stage $s+1$, let $r_A(e,s)$ and $r_B(e,s)$ be the amount of σ_s and τ_s that we have to preserve to save the work done until now for R_e, and let

$$R_A(e,s) = \max_{i<e} \ r_A(i,s) \qquad R_B(e,s) = \max_{i<e} \ r_B(i,s).$$

We then proceed as follows:

- *if s is even and $s = \langle e, x, m\rangle$*
  This is one of infinitely many chances we are given to satisfy $C_{e,x}$. For each $n \leq s$ such that $C_{e,x,n}$ is not yet satisfied (i.e. σ_s does not give value 1 to any element of type $\langle n, w\rangle$ of the $\langle e, x\rangle$-th column), see if $(\forall y \leq n)(\exists z \leq s)Q(x, y, z, e)$, and $\{e\}^E(n)$ converges in at most s steps. If so, put $\langle n, \{e\}^E(n)\rangle$ in the $\langle e, x\rangle$-th columns of both A and B.

- *if s is odd*
  Look for the smallest $e \leq s$ such that R_e is not yet satisfied but it can be satisfied at this stage, i.e. there are strings σ and τ of length $\leq s$, e-splitting on some x with computations converging in at most s steps, and such that:

  1. σ extends $\sigma_s[R_A(e, s)]$ and is compatible with the columns of σ_s with index $\leq e$
  2. τ extends $\tau_s[R_B(e, s)]$ and is compatible with the columns of τ_s with index $\leq e$.

  If such an e exists, then choose the smallest strings as above relative to it, and let $\sigma_{s+1} = \sigma$ and $\tau_{s+1} = \tau$. Moreover, let

  $$r_A(i, s+1) = \begin{cases} r_A(i, s) & \text{if } i < e \\ |\sigma| & \text{if } i = e \\ 0 & \text{otherwise,} \end{cases}$$

  and similarly for $r_B(i, s+1)$.

  Otherwise, everything is left at stage $s + 1$ as it was at stage s.

By the usual finite injury argument,

$$\lim_{s\to\infty} r_A(e, s) < \infty \quad \text{and} \quad \lim_{s\to\infty} r_B(e, s) < \infty,$$

and

$$\lim_{s\to\infty} \sigma_s(x) \quad \text{and} \quad \lim_{s\to\infty} \tau_s(x)$$

exist, for every x. Since the construction is recursive in E, by the Limit Lemma we have $A, B \leq_T E'$ and, by lowness of E, $A, B \leq_T \mathcal{K}$.

Since each requirement R_e can interfere only with columns with index $\geq e$ and its satisfaction only requires finite action, with at most finitely many exceptions we put elements into column $\langle e, x\rangle$ just to satisfy $C_{e,x}$. Thus if $\langle n, w\rangle$ is on the column, then $w = \{e\}^E(n)$ by construction, with at most finitely many exceptions. On the other hand, if $(\forall y)(\exists z)Q(x, y, z, e)$, then we do put an element $\langle n, w\rangle$ for each n in column $\langle e, x\rangle$. To compute $\{e\}^E(n)$, it is then enough to look for an element $\langle n, w\rangle$ on column $\langle e, x\rangle$ of A (or B), since $\{e\}^E(n) = w$ except for finitely many n, for which $\{e\}^E(n)$ can be given for free. Thus $\{e\}^E \leq_T A, B$ and C_e is satisfied.

It remains to prove by induction on e that R_e is satisfied. Choose s_0 large enough so that the requirements R_i with $i < e$ have been permanently satisfied before stage s_0, if ever, and such that the restraint functions $R_A(e, s)$ and $R_B(e, s)$ have reached their final values. In particular, $\sigma_s[R_A(e, s)] \subseteq A$ and $\tau_s[R_B(e, s)] \subseteq B$ for each $s \geq s_0$. Suppose the hypothesis of R_e is satisfied, i.e. $\{e\}^A \simeq \{e\}^B \simeq C$ for some C. Given x, $\{e\}^A(x)$ and $\{e\}^B(x)$ are then both defined, and thus there is a stage $s \geq s_0$ for which there exist $\sigma \supseteq \sigma_s[R_A(e, s)]$ compatible with the first e columns of σ_s, as well as a $\tau \supseteq \tau_s[R_B(e, s)]$ compatible with the first e columns of τ_s, such that $\{e\}^\sigma$ and $\{e\}^\tau$ both converge on x in at most s steps. Since $\{e\}^A \simeq \{e\}^B$ by hypothesis, such strings cannot e-split, otherwise they would be picked up during the construction (since R_e has highest priority after stage s_0). Thus they give the right value of $C(x)$. To find such a stage is recursive in E because such is the construction, but we do not really need all of E. Knowledge of the first e columns, and hence of the $\{m\}^E$ for $m \leq e$ and $m \in I$, is sufficient (for the compatibility condition). Thus C is recursive in $\bigoplus_{m \leq e \wedge m \in I} \{m\}^E$ and R_e is satisfied. $\Box$

With a little effort and by standard techniques we can obtain the following improvement which, according to the discussion at the beginning, is almost the best one can hope for.

Theorem XI.3.18 (Shore [1981a]) *If $\mathcal{I}$ is an ideal bounded below a degree $\boldsymbol{e} < \boldsymbol{0'}$ and Σ^0_3 in it, then it has an exact pair below $\boldsymbol{0'}$.*

Proof. The construction of the previous result was recursive in E and produced an exact pair recursive in E'. This is not enough in general, since we actually want the exact pair recursive in $\mathcal{K}$. The main new ingredient now is the use of permitting below $\mathcal{K}$ (XI.2.8), which is applicable because by hypothesis $\mathcal{K}$ is r.e. but not recursive in E.

Fix a one-one recursive function f with range $\mathcal{K}$. To obtain $A, B \leq_T \mathcal{K}$ we will require that, at any stage s,

$$\sigma_{s+1} \supseteq \sigma_s[f(s)] \quad \text{and} \quad \tau_{s+1} \supseteq \tau_s[f(s)].$$

The coding requirements might be difficult to satisfy if we insisted on coding $\{e\}^E$ by putting $\langle n, \{e\}^E(n)\rangle$ in the $\langle e, x\rangle$-th column, since this might interfere with the permitting strategy (when $\langle n, \{e\}^E(n)\rangle$ is not permitted at stage $s+1$). But a simple variation will work. Instead of using the fixed element $\langle n, \{e\}^E(n)\rangle$ to code $\{e\}^E(n)$, we use an element $\langle v, n, \{e\}^E(n)\rangle$. By so doing we can thus *choose* v, and thus push $\langle v, n\{e\}^E(n)\rangle$ outside the part we have to preserve for the sake of permitting. Thus the revised coding strategy is not affected by permitting.

The restriction imposed by permitting does however affect the part of the construction relative to the minimality requirements, since it makes it more difficult for a string to change. But as in any typical permitting argument one shows that if some requirement R_e were not satisfied, then $\mathcal{K}$ would be recursive in E, contradicting the hypothesis. $\Box$

The only possible improvement of the last result would be by changing Σ_3^0 in $\boldsymbol{e}$ into straight Σ_4^0. Shore [1981a] has proved that it can be done when $\boldsymbol{e}$ is low$_2$, i.e. $\boldsymbol{e}'' = \mathbf{0}''$, but we do not know whether it can be done in general.

XI.4 Minimal Degrees

We now look at analogues of the results of Section V.5, in particular the method of trees and its main application, the construction of minimal degrees.

Methodology $\star$

The notion of a tree was introduced in V.5.1. Recall that a **partial recursive tree** is a partial recursive function T on strings such that

1. $T(\sigma)\downarrow \wedge\, \tau \subseteq \sigma \;\Rightarrow\; T(\tau)\downarrow \wedge\, T(\tau) \subseteq T(\sigma)$
2. if one of $T(\sigma * 0), T(\sigma * 1)$ is defined, both are defined and incompatible.

In particular, the range of a partial recursive tree is an r.e. set of strings.

The simplest construction of a minimal degree was given in V.5.11, where we used total recursive trees (with no particular uniformities, unlike those of Section V.6) which were either e-splitting or without e-splittings, for some e. We have already argued (see the comments after V.5.11) that one cannot use total recursive trees in the construction of minimal degrees below $\mathbf{0}'$. The next result is another proof of the same fact.

Proposition XI.4.1 (Posner [1981]) *Let $A \in \bigcap_{e\in\omega} T_e$, where T_e is a partial recursive tree which is either e-splitting or without e-splittings. If $A \leq_T \mathcal{K}$, then A is the only infinite branch of some T_e.*

Proof. Let $\{\sigma_s\}_{s\in\omega}$ be a recursive approximation to A. There is an e such that

$$\{e\}^{\sigma}(x) \simeq \sigma(x) \;\Leftrightarrow\; (\exists s)(\sigma \subseteq \sigma_s).$$

First note that there are e-splittings on T_e. Otherwise A would be recursive (contradicting V.5.12.c), since to compute $A(x)$ it would be enough to search for σ on T_e such that $\{e\}^\sigma(x)\downarrow$, which exists because A is on T_e.

Since there are e-splittings on T_e, by hypothesis T_e is e-splitting. Now consider $T_e(\sigma * 0)$ and $T_e(\sigma * 1)$, for any σ. If they were both defined, then they would split on some argument, and thus $\{e\}^{T(\sigma * i)}$ $(i = 0, 1)$ would be defined for that argument. By definition this implies that $T_e(\sigma * i) \subseteq \sigma_{s_i}$, for some s_i. Since $T_e(\sigma * 0)$ and $T_e(\sigma * 1)$ are incompatible, $s_0 \neq s_1$. But $\lim_{s\to\infty} \sigma_s(x)$ exists for every x, and thus one of $T_e(\sigma * 0)$ and $T_e(\sigma * 1)$ can have only finitely many extensions on T_e. $\square$

Thus, if we want to use the method of V.5.11 to build minimal degrees below $\mathbf{0}'$ we must drop total trees and use partial recursive trees instead. On the other hand, we now show that the use of partial recursive trees that are either e-splitting of without e-splittings is almost necessary, in the sense made precise by the next definition and proposition (which are weak analogues of, respectively, V.5.8 and the converse of V.5.9).

Definition XI.4.2 *A partial recursive tree T (i.e. an r.e. set of strings) is:*

1. **weakly without *e*-splittings** *if, for some partial recursive $\varphi : T \to \omega$, whenever σ and τ are on T, they do not e-split below* $\min\{\varphi(\sigma), \varphi(\tau)\}$
2. **weakly *e*-splitting** *if, for some partial recursive $\varphi, \psi : T \to \omega$, whenever σ and τ on T are incompatible below* $\min\{\varphi(\sigma), \varphi(\tau)\}$*, they e-split below* $\min\{\psi(\sigma), \psi(\tau)\}$.

Proposition XI.4.3 (Chong [1979]) *Let $A \leq_T \mathcal{K}$ be of minimal degree and let $\{e\}^A$ be total. Then there is a partial recursive tree T such that A is on T and*

1. *if $\{e\}^A$ is recursive, T is weakly without e-splittings*
2. *if $A \leq_T \{e\}^A$, T is weakly e-splitting.*

Moreover, T is not trivial, i.e. for φ and ψ as in XI.4.2

$$\sup_{\sigma \subseteq A \,\wedge\, \sigma \in T} \varphi(\sigma) = \omega \qquad \text{and} \qquad \sup_{\sigma \subseteq A \,\wedge\, \sigma \in T} \psi(\sigma) = \omega.$$

Proof. Suppose $\{e\}^A$ is total recursive. The idea is to start the tree T by putting up all possible minimal strings σ such that $\{e\}^\sigma$ and $\{e\}^A$ agree on 0, and to let $\varphi(\sigma) = 0$: no two such strings σ_1 and σ_2 e-split on 0, since $\{e\}^{\sigma_1}$ and $\{e\}^{\sigma_2}$ both agree with $\{e\}^A$ on 0.

The problem is that we cannot check effectively which strings satisfy the condition because, while $\{e\}^A$ is total recursive and can be freely used, $\{e\}^\sigma$ could instead be undefined on 0 for some σ: we thus have to recursively bound the computations, by choosing an appropriate recursive function f, and putting up only the minimal strings σ such that $\{e\}^\sigma_{f(0)}$ and $\{e\}^A$ agree on 0. At the next stage we add to T the minimal strings σ such that $\{e\}^\sigma_{f(1)}$ and $\{e\}^A$ agree up to 1, and let $\varphi(\sigma) = 1$ for all such strings, and so on.

We now have to choose the recursive function f in such a way that A is on T, i.e. such that infinitely often a new initial segment of A is put up on T. For this purpose, we consider the function

$$h(x) = \mu s.\,(\exists \sigma)(x \leq |\sigma| \leq s \ \wedge\ \sigma \subseteq A \ \wedge\ \{e\}^\sigma_s, \{e\}^A \text{ agree up to } x).$$

Since A is not high by XI.2.15, no function recursive in A dominates every total recursive function by XI.1.2; since h is recursive in A, there exists a total recursive function f that is not dominated by h, i.e. such that $h(x) < f(x)$ for infinitely many x, and we can use this function in the construction of T.

Suppose now that $A \leq_T \{e\}^A$, and hence that $A \simeq \{i\}^{\{e\}^A}$ for some i. The idea is to start the tree T by putting up all possible minimal strings σ such that $\{i\}^{\{e\}^\sigma}$ agrees with σ on 0, and to let $\varphi(\sigma) = 0$, and $\psi(\sigma)$ be the use function of $\{i\}^{\{e\}^\sigma}$, i.e. the maximum x such that $\{e\}^\sigma(x)$ was used to compute $\{i\}^{\{e\}^\sigma}$ on 0: if two such strings σ_1 and σ_2 are incompatible on 0, then so are the respective computations $\{i\}^{\{e\}^{\sigma_1}}$ and $\{i\}^{\{e\}^{\sigma_2}}$, which agree with them, and hence their oracles $\{e\}^{\sigma_1}$ and $\{e\}^{\sigma_2}$ must differ below the use, which means that the two strings σ_1 and σ_2 e-split below the use.

As above, the problem is that we cannot check effectively which strings satisfy the condition: we thus have to recursively bound the computations, by choosing an appropriate recursive function f, and putting up only the minimal strings σ such that $\{i\}^{\{e\}^\sigma}_{f(0)}$ agrees with σ on 0. At the next stage we add to T the minimal strings σ such that $\{i\}^{\{e\}^\sigma}_{f(1)}$ agrees with σ up to 1, and let $\varphi(\sigma) = 1$ for all such strings, and $\psi(\sigma)$ be the maximum x such that $\{e\}^\sigma(x)$ was used to compute $\{i\}^{\{e\}^\sigma}$ up to 1, and so on.

Again, we have to choose the recursive function f in such a way that A is on T, i.e. such that infinitely often a new initial segment of A is put up on T. For this purpose, we consider the function

$$h(x) = \mu s.\,(\exists \sigma)(x \leq |\sigma| \leq s \ \wedge\ \sigma \subseteq A \ \wedge\ \{i\}^{\{e\}^\sigma}_s, \sigma \text{ agree up to } x),$$

and proceed as above. □

Besides using total trees, the construction of a minimal degree (below $\mathbf{0}''$) in V.5.11 amounted to the definition of a function $h \leq_T \emptyset''$ with the property that

$h(e)$ was defined at stage e as an index of a tree T_e which was either e-splitting or without e-splittings. This simple procedure of satisfying each condition for minimality once and for all cannot be used to obtain a minimal degree below $\mathbf{0}'$.

Proposition XI.4.4 (Posner [1981]) *Let $\{T_e\}_{e\in\omega}$ be a sequence of partial recursive trees such that T_e is either e-splitting or without e-splittings. If $\bigcap_{e\in\omega} T_e \neq \emptyset$, then there is no function $h \leq_T \mathcal{K}$ such that $h(e)$ is an index of T_e.*

Proof. Let f be a recursive function such that

$$\{f(e)\}^\sigma(x) \simeq \sigma(x) \;\Leftrightarrow\; (\forall y < x)(\{e\}^\sigma(y)\downarrow).$$

Fix $A \in \bigcap_{e\in\omega} T_e$, which exists by hypothesis. We prove that

$$\{e\}^A \text{ total } \Leftrightarrow\; T_{f(e)} \text{ is } f(e)\text{-splitting.}$$

- If $T_{f(e)}$ is $f(e)$-splitting, to show that $\{e\}^A(y)\downarrow$ consider σ such that $T_{f(e)}(\sigma) \subseteq A$, and whose length is greater than y. One of $T_{f(e)}(\sigma * 0)$ and $T_{f(e)}(\sigma * 1)$ is contained in A, say $T_{f(e)}(\sigma * i)$. Since $T_{f(e)}$ is $f(e)$-splitting, $T_{f(e)}(\sigma * 0)$ and $T_{f(e)}(\sigma * 1)$ $f(e)$-split on some argument x. In particular, $\{f(e)\}^{T_{f(e)}(\sigma * i)}(x)\downarrow$, hence $\{e\}^{T_{f(e)}(\sigma * i)}$ converges on all arguments less that x. Moreover, $y < x$ because $T_{f(e)}(\sigma * 0)$ and $T_{f(e)}(\sigma * 1)$ agree up to $|T_{f(e)}(\sigma)| > y$, but differ on x.

- If $\{e\}^A$ is total, to show that $T_{f(e)}$ is e-splitting it is enough to show that there are $f(e)$-splittings on it. This is so, otherwise A would be recursive (contradicting V.5.12.c), since to compute $A(x)$ it would be enough to look for any σ on $T_{f(e)}$ such that $\{f(e)\}^\sigma(x)\downarrow$.

But '$\{e\}^A$ is total' has the same degree as A'', and so to compute A'' we only need to know if $T_{f(e)}$ is $f(e)$-splitting. But since $T_{f(e)}$ is either e-splitting or without e-splitting, it is enough to see if $T_{f(e)}(0)$ and $T_{f(e)}(1)$ $f(e)$-split. This can be done recursively in $\mathcal{K}$, once we know how to compute $T_{f(e)}$. Suppose h gives us an index of $T_{f(e)}$, i.e. it tells us how to compute it. If $h \leq_T \mathcal{K}$, we can compute A'' recursively in $\mathcal{K}$, which is a contradiction. □

Minimal degrees below $\mathbf{0}'$

If we want to build a minimal degree below $\mathbf{0}'$ the results of the previous subsection do not leave us much choice. We must use *partial* recursive trees which are either e-splitting or without e-splittings. And the construction cannot satisfy each requirement for minimality once and for all. The next proof is thus almost forced on us.

Theorem XI.4.5 (Sacks 1961a]) *There exists a minimal degree below* $\mathbf{0}'$.

Proof. The basic ingredients of the construction of a minimal degree (V.5.11) are diagonalizing against every recursive function, and constructing trees which are either e-splitting or without e-splittings. The first step (which, by V.5.12.c, is avoidable) requires only one-quantifier questions, and it is thus recursive in $\mathcal{K}$, but the second apparently requires two-quantifier questions. The idea of the present proof is to mimic the second step as far as possible, by asking only one-quantifier questions.

Suppose we are given T and e and we want to build $Q \subseteq T$. Instead of asking if every $\sigma \in T$ has e-splitting extensions on T, we pretend this is so and define Q as the **full e-splitting subtree** of T. More precisely, $Q(\emptyset) = T(\emptyset)$, and $Q(\sigma * 0), Q(\sigma * 1)$ are the smallest e-splitting extensions of $Q(\sigma)$ on T. Of course, it may very well be that T is not as we pretended and that, for some σ, $Q(\sigma)$ does not have e-splitting extensions on T. This makes $Q(\sigma * i)$ $(i = 0, 1)$ undefined and Q partial, even if T was total. If at a later stage of the construction we ask if a given string σ we are interested in (namely, a beginning of the set we are constructing) has incompatible extensions on Q (this being a one-quantifier question), and we find out it does not, then we discover that our assumption on T was a false one. We will then change our mind about Q, taking the full subtree of T above σ instead. This is because, by construction, the lack of incompatible extensions of σ on Q means that σ has no e-splitting extensions on T, and thus Q will be a tree without e-splittings.

We build $A \leq_T \mathcal{K}$ of minimal degree by initial segments σ_s, so that the trees play only an auxiliary role. The function $g(s)$ tells us how many trees we continue to believe are good at the end of stage s. T_e^s is what we think T_e is at stage s.

The construction is as follows. We start with $\sigma_0 = \emptyset$, $g(0) = 1$, T_0^0 the identity tree, and T_1^0 the full 0-splitting subtree of T_0^0. At step $s+1$, we have σ_s and

$$T_0^s \supseteq T_1^s \supseteq \cdots \supseteq T_{g(s)}^s.$$

Let i_0 be the greatest $i \leq g(s)$ such that σ_s has a proper extension on T_i^s, which exists because $T_0^s = T_0^0$ is always the identity tree. If there is one extension, there are two incomparable ones, by the definition of a tree. Let σ_{s+1} be a proper extension of σ_s on $T_{i_0}^s$ incompatible with the recursive function $\{s\}$, so that $A \not\simeq \{s\}$. Also, $T_i^{s+1} = T_i^s$ for $i \leq i_0$, since there is no need to change these trees. If $i_0 = g(s)$, then we still believe that all our trees at step s were good and we simply add one more to keep the construction going. Thus we let

$$T_{i_0+1}^{s+1} = i_0\text{-splitting subtree of } T_{i_0}^{s+1}.$$

If $i_0 < g(s)$, then we discovered that $T^s_{i_0+1}$ is not i_0-splitting above σ_s, since σ_s has no proper extensions on it. Thus we let

$$T^{s+1}_{i_0+1} = \text{ full subtree of } T^{s+1}_{i_0} \text{ above } \sigma_{s+1}.$$

All the other T^s_i for $i > i_0 + 1$ are dropped, since we built them as subtrees of $T^s_{i_0+1}$, which we discovered was a wrong guess. In both cases, $g(s+1) = i_0 + 1$.

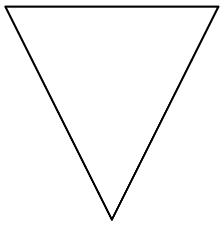
Note that $\lim_{s\to\infty} T^s_e = T_e$ exists because T^s_e can change at most $2^e - 1$ times. Indeed, T^s_0 never changes, T^s_1 can change only once, T^s_2 can change once for each choice of T^s_1, hence 3 times, and so on.

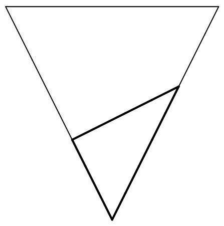
By construction T_{e+1} is either e-splitting or has no e-splittings above σ_s, for any s such that $T^s_{e+1} = T_{e+1}$. Since $A = \bigcup_{s\in\omega} \sigma_s$ is on every T_e, A has minimal degree. Finally, $A \leq_T \mathcal{K}$ because the construction only asks one-quantifier questions, and the approximations σ_s to A are never changed, i.e. A is built by a $\mathbf{0}'$-oracle construction. $\square$

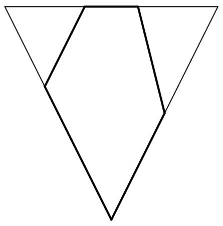
If we want to know T_e itself, or to find an s such that $T^s_e = T_e$, then we still need a $\mathbf{0}''$-oracle. But A itself was built recursively in $\mathbf{0}'$ only.

By V.5.12.c, there is no need to diagonalize. But even if we do not, we still have to define σ_{s+1} as a proper extension of σ_s, otherwise at the end $\bigcup_{s\in\omega} \sigma_s$ would not be total.

The basic idea of the construction was the following. Given T and e, let Q be the e-splitting subtree of T. Three possibilities may arise:


(1)


(2)


(3)

1. *Q is a total tree*
   If T does not change, then Q will not change either, since every branch of it satisfies the e-th minimality requirement.

2. *Q has no infinite branch*
   Even if T does not change, then Q is bound to change, since the construction produces an infinite branch that cannot be on Q.

3. *Q has some infinite branches, but it is not total*
   Even if T does not change, we cannot tell before hand whether or not Q will change, because everything depends on the following steps of the construction. If A is pushed through the boundary of Q, then we will drop Q. If A remains inside Q, then Q will be kept.

More comments on the construction are given at the beginning of the next subsection.

Muchnick [1956] proved that there is no minimal r.e. degree (X.2.8). This provides another two-quantifier elementary difference between the two structures $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ and $\boldsymbol{\mathcal{R}}$.

Exercises XI.4.6 a) *Given a nontrivial degree below $\mathbf{0}'$, there is a minimal degree below $\mathbf{0}'$ incomparable with it.* (Shoenfield [1966]) (Hint: build a tree recursive in $\mathbf{0}'$, all of whose branches have minimal degree. Then apply the second proof of XI.3.6.)

b) *There is a minimal degree below $\mathbf{0}'$ incomparable with every nontrivial r.e. degree.* (Sasso [1970]) (Hint: see part a) and XI.3.7.a.)

c) *There are infinitely many minimal degrees below $\mathbf{0}'$.* (Fukuyama [1968])

d) *There exists a completely autoreducible degree below $\mathbf{0}'$.* (Jockusch and Paterson [1976]) (Hint: apply the technique of XI.4.5 to doubly e-splitting trees, see V.5.15.d.)

Exercises XI.4.7 Jumps of minimal degrees below $\mathbf{0}'$. Recall that, by XI.2.15, every minimal degree below $\mathbf{0}'$ is in $\mathbf{L}_2$.

a) *There are minimal degrees in $\mathbf{L}_1$.* (Yates [1970a]) (Hint: by XI.4.9, since there are nonrecursive low r.e. degrees.) The hint relies on a result proved by the full approximation method. A complicated $\mathbf{0}'$-oracle construction of a minimal degree in $\mathbf{L}_1$ is given in Lerman [1983].

b) *There are minimal degrees in $\mathbf{L}_2 - \mathbf{L}_1$.* (Sasso [1974], Cooper, Epstein) (Hint: it is enough to build a minimal degree below $\mathbf{0}'$ that is not low. We modify V.5.14.a by applying the guessing technique of XI.4.5. When satisfying the nonlowness requirements, we first guess that there is no $\tau \in T$ such that $\{e\}^{\tau \oplus \mathcal{K}}(a) \simeq 0$. If we later find an initial segment τ of A for which this holds, then we change the tree.)

c) *Every degree above $\mathbf{0}''$ and r.e. in it is the double jump of a minimal degree below $\mathbf{0}'$.* (Jockusch and Posner [1978]) (Hint: any such degree is the jump of a set C of degree between $\mathbf{0}'$ and $\mathbf{0}''$, by relativization of the Shoenfield Jump Theorem. We use the technique of XI.4.5 to build a tree recursive in $\mathcal{K}$ and all of whose branches are of minimal degrees. By following the branch determined by C, we find a set A of minimal degree such that $C \equiv_T A \oplus \mathcal{K}$. By XI.2.15, the degree of A is in $\mathbf{GL}_2$, i.e. $A'' \equiv_T (A \oplus \mathcal{K})' \equiv_T C'$. Thus C has been chosen with the desired jump.)

Necessary conditions for a degree to be the jump of a minimal degree below $\mathbf{0}'$ are to be r.e. in and above $\mathbf{0}'$ and to have jump $\mathbf{0}''$ (since minimal degrees below $\mathbf{0}'$ are in $\mathbf{L}_2$). Cooper [1996] and Downey, Lempp and Shore [1996] have shown that these conditions are not sufficient. Cooper [1996] has stated necessary and sufficient conditions.

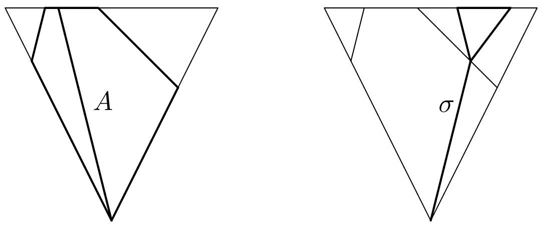
Full approximation arguments ⋆

As we have already seen in Section 2, turning a $\mathbf{0}'$-oracle construction into a full approximation argument introduces complications, but it may have interesting consequences, such as pushing the constructed set below a nonrecursive r.e. degree.

The original construction of a minimal degree (V.5.11) was recursive in $\mathbf{0}''$. We have just turned it into a construction recursive in $\mathbf{0}'$ and are now looking for a further improvement, namely a recursive construction. To set out the difficulties, let us examine the basic step of XI.4.5 more fully.

Note that if T is total and recursive, then we can build a total tree Q (not partial recursive, but) recursive in $\mathbf{0}'$ which has a partial recursive subtree Q^* such that if T does not change, then Q^* is the final tree relative to the e-th minimality requirement. Simply start with $Q(\emptyset) = T(\emptyset)$ and, given $Q(\sigma)$, let $Q(\sigma * 0)$ and $Q(\sigma * 1)$ be the smallest e-splitting extensions of $Q(\sigma)$ on T if they exist, and the smallest incompatible extensions of $Q(\sigma)$ on T otherwise.

The definition of Q asks one-quantifier questions about T and gives a complete picture of every possibility that might arise in the construction. Basically, it puts up as many e-splittings as possible, and then copies T. Given any branch A of Q, then:



(1) (2)

1. if A is a branch of the e-splitting part of Q, let Q^* be the e-splitting subtree of T

2. if A passes through the boundary of the e-splitting part of Q, let $\sigma \subseteq A$ be the string on the boundary and let Q^* be the full subtree of T above σ.

We might think of iterating the construction just given by starting with T_0 as the identity tree and letting T_1 be the tree constructed of two parts: a 0-splitting bottom one, and a top one without 0-splittings. Then we could define T_2 by putting up as many 1-splittings as possible in each part, and copying T_1. This would produce a tree T_2 consisting of four parts: a 0-splitting, 1-splitting part; a 0-splitting part without 1-splittings; a 1-splitting part without 0-splittings; and finally a part without 0-splittings and without 1-splittings. Then T_3 would consist of 8 parts, and so on. By always letting $T_{e+1}(\emptyset) = T_e(0)$, we could obtain

$$A = \bigcup_{e\in\omega} T_e(\emptyset) \in \bigcap_{e\in\omega} T_e.$$

Then A would be of minimal degree, since it would be possible to obtain partial recursive trees $T^*_{e+1} \subseteq T_{e+1}$ that are either e-splitting or without e-splittings, and such that A is on T^*_{e+1}. The point is that by doing so we would not obtain $A \leq_T \mathcal{K}$, because T_{e+1} is only recursive in the jump of T_e.

What we need is a constructivization of this procedure that would allow us to obtain each T_e recursively in $\mathbf{0}'$. Then A itself will be recursive in $\mathbf{0}'$. This is the idea behind the full approximation construction given below. The method used is the same adopted in III.4.18 to turn the noneffective construction of a cohesive set in III.4.16 into a recursive construction of a maximal set.

Proposition XI.4.8 (Yates [1970a], Cooper [1972a]) *Minimal degrees below $\mathbf{0}'$ can be built by full approximation.*

Proof. We build total recursive trees T^s_e in a uniform way and with the property that, given $T_{e+1} = \lim_{s\to\infty} T^s_{e+1}$, and $A = \bigcup_{s\in\omega} T_e(\emptyset)$, there is a partial recursive tree $T^*_{e+1} \subseteq T_{e+1}$ which is either e-splitting or without e-splittings, and such that A is on T^*_{e+1}. By V.5.12.c, A is also nonrecursive and thus it has minimal degree. Moreover, $A \leq_T \mathcal{K}$ by the Limit Lemma, since $A = \lim_{s\in\omega} T^s_s(\emptyset)$.

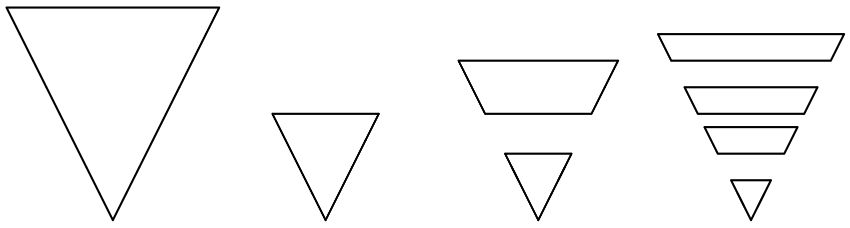
The idea is to have

$$T_{e+1} \subseteq T_e \subseteq \cdots \subseteq T_0 = \text{ identity tree,}$$

with each T_e made of 2^e parts, ordered upwards by decreasing e-states, where the **e-state** of σ is

$$\sum\{2^{e-i} : i \leq e \ \wedge\ (\exists\tau)(\sigma \text{ and } \tau \ i\text{-split})\}.$$

Since we want T^s_e to be recursive, at stage s we only look for splittings of length bounded by s. We thus let the e-state of σ at stage s be defined as above only if $|\sigma|, |\tau| \leq s$, and let it be 0 otherwise (and also if σ does not i-split with any τ of length $\leq s$, for all $i \leq e$).



The construction is as follows. At stage s we define T_e^s by induction, first on e and then on the length of σ, by trying to maximize e-states:

1. $T_0^s = T_0$ is the identity tree (we do not try to approximate it, because it is already recursive)
2. $T_{e+1}^s(\emptyset) = T_e^s(0)$
3. given $T_{e+1}^s(\sigma)$, let $T_{e+1}^s(\sigma * 0)$ and $T_{e+1}^s(\sigma * 1)$ be the smallest incompatible extensions of $T_{e+1}^s(\sigma)$ on T_e^s with maximal e-state at stage s (for the same e).

For strings of length greater than s, the definition of T_{e+1}^s simply copies T_e^s above the strings already defined, since for these strings the e-state is always 0. Also, at most s trees are defined nontrivially at stage s, because condition 2 causes $T_{s+1}^s(\emptyset)$ to be of length greater than s.

By induction on e and $|\sigma|$, $\lim_{s\to\infty} T_e^s(\sigma)$ exists. Indeed, when T_e^s and $T_{e+1}^s(\sigma)$ have both reached their limits, then $T_{e+1}^s(\sigma * 0)$ and $T_{e+1}^s(\sigma * 1)$ can only change to reach a higher state, so only finitely often. Then $T_e = \lim_{s\to\infty} T_e^s$ exists and is recursive in $\mathbf{0}'$. Thus $A = \bigcup_{e\in\omega} T_e(\emptyset)$ is well-defined, because

$$T_e(\emptyset) \subseteq T_e(0) = T_{e+1}(\emptyset),$$

and $A \leq_T \mathcal{K}$ because $A = \lim_{s\to\infty} T_s^s(\emptyset)$.

Since the e-states decrease monotonically, for any branch B of T_{e+1} there is a string σ_B such that $T_{e+1}(\sigma_B) \subseteq B$, and an e-state s_B such that for all $\tau \supseteq \sigma_B$, if $T_{e+1}(\tau * i) \subseteq B$ ($i = 0$ or 1), then both $T_{e+1}(\tau * 0)$ and $T_{e+1}(\tau * 1)$ have e-state s_B. We call s_B the final e-state of B.

Recall that $A = \bigcup_{e\in\omega} T_e(\emptyset)$ and let T_{e+1}^* be the subtree of T_{e+1} above $T_{e+1}(\sigma_A)$, consisting of all branches (infinite or not) with final e-state s_A. A is clearly on T_{e+1}^*. By the properties of the e-states, T_{e+1}^* is partial recursive and either e-splitting or without e-splittings. Indeed, it is impossible for $T_{e+1}(\sigma)$ to have e-splitting extensions if $T_{e+1}(\sigma * 0)$ and $T_{e+1}(\sigma * 1)$ do not e-split, since otherwise the construction would sooner or later pick up the e-splitting, and define $T_{e+1}(\sigma * 0)$ and $T_{e+1}(\sigma * 1)$ as e-splitting. Hence A has minimal degree. $\square$

The main difference between the constructions of XI.4.5 and XI.4.8 is that in the former the trees are approximated globally (as trees) and in the latter locally (string by string). Thus in the $\mathbf{0}'$-oracle construction a tree can be injured, i.e. caused to change by some tree with lower index, only finitely many times, while in the full approximation construction a tree can be injured infinitely many times, because each injury might be produced by a different string, and the fact that each string comes individually to a limit after finitely many stages does not help, since the whole tree does not. What cannot happen is that we injure a tree infinitely many times along a single branch, because in this case the e-states control the injuries.

Permitting below r.e. degrees $\star$

We now add permitting below r.e. degrees to the full approximation construction of a minimal degree.

Theorem XI.4.9 (Yates [1970a]) *Every nonzero r.e. degree bounds a minimal degree.*

Proof. Let C be a nonrecursive r.e. set and let f be a one-one recursive function with range C. We keep the same notation as in XI.4.8 and add permitting (see XI.2.8). If $A = \lim_{s\to\infty} \sigma_s$, we want to ensure $\sigma_{s+1} \supseteq \sigma_s[f(s)]$. Since σ_s is defined in terms of trees (i.e. $\sigma_s = T_s^s(\emptyset)$), to achieve this we require that no change be allowed on any tree at any stage unless it is permitted. So let:

1. $T_0^s = T_s^0 = T_0$ is the identity tree
2. $T_{e+1}^{s+1}(\emptyset)$ is $T_e^{s+1}(0)$ if this is compatible with $\sigma_s[f(s)]$, and $T_e^{s+1}(1)$ otherwise.
3. given $T_{e+1}^{s+1}(\sigma)$, consider the smallest incompatible extensions σ^0 and σ^1 of it on T_e^{s+1}, and see if on T_e^{s+1} there are permitted (i.e. $\supseteq \sigma_s[f(s)]$) incompatible extensions with higher e-state at stage $s+1$. If not, let $T_{e+1}^{s+1}(\sigma * i) = \sigma^i$. Otherwise, let $T_{e+1}^{s+1}(\sigma * 0)$ and $T_{e+1}^{s+1}(\sigma * 1)$ be the smallest incompatible extensions of $T_{e+1}^{s+1}(\sigma)$ with e-state at stage $s+1$ maximal among the e-states of permitted extensions.

Extreme cases that might arise during the construction are:

- No extension of $T_{e+1}^{s+1}(\sigma)$ is permitted (e.g. $\sigma_s[f(s)]$ and $T_{e+1}^{s+1}(\sigma)$ are incompatible), so we copy T_e^{s+1} above it.

- Every extension of $T^{s+1}_{e+1}(\sigma)$ is permitted (e.g. $\sigma_s[f(s)] \subseteq T^{s+1}_{e+1}(\sigma)$), so the construction above $T^{s+1}_{e+1}(\sigma)$ proceeds as in XI.4.8, as if it were without permitting.

The changes in the construction ensure that $\sigma_{s+1} \supseteq \sigma_s[f(s)]$. Indeed, by definition, $\sigma_{s+1} = T^{s+1}_{s+1}(\emptyset) \supseteq T^{s+1}_s(\emptyset)$. If $T^s_s(\emptyset) = T^{s+1}_s(\emptyset)$, then there is nothing to prove, since $\sigma_{s+1} = \sigma_s$. If $T^s_s(\emptyset) \neq T^{s+1}_s(\emptyset)$, then a change was possible only because it was permitted, so $T^{s+1}_s(\emptyset) \supseteq \sigma_s[f(s)]$. It follows that $A \leq_T C$.

Since permitting makes it harder for trees to change, we still have that $\lim_{s\to\infty} T^s_e(\sigma)$ exists, and that each branch has a final e-state. So we can define T^*_{e+1} as before, and have A on T^*_{e+1}, and T^*_{e+1} partial recursive. It remains to prove by induction on e that T^*_{e+1} is either e-splitting or without e-splittings. Suppose T^*_{e+1} is not e-splitting, but every beginning of A has e-splitting extensions on it (otherwise only finitely many have, so we could take $\sigma \subseteq A$ with no e-splitting extension, and have a partial recursive tree without e-splittings and with A on it, and this would be enough for the e-th requirement for minimality).

We show as in XI.2.10 that C is recursive, contradicting the hypothesis. Consider σ_A as in XI.4.8 and choose s_0 large enough so that $T^{s_0}_{e+1}$ has settled down to level σ_A. Given x, find $s \geq s_0$ such that, for some $\sigma \supseteq \sigma_A$ with $|\sigma| \leq s$, $\sigma_s[x] \subseteq T^s_{e+1}(\sigma)$, and $T^s_{e+1}(\sigma * 0)$ and $T^s_{e+1}(\sigma * 1)$ have e-state s_A, and they are not e-splitting, but there are e-splitting extensions of $T^s_{e+1}(\sigma)$ on T^s_{e+1} of length $\leq s$.

Such an s exists by the hypothesis because, for a sufficiently large s, $\sigma_s[x]$ is contained in A. Note that

$$t \geq s \;\Rightarrow\; \sigma_t[x] = \sigma_s[x],$$

by induction on $t \geq s$. Indeed, by construction $\sigma_{t+1} \supseteq \sigma_t[f(t)]$. Moreover, $f(t) \geq x$. Otherwise, by the induction hypothesis that $\sigma_t[x] = \sigma_s[x]$, the e-splitting extensions of $\sigma_s[x]$ would also be above $\sigma_t[f(t)]$, and thus they would be chosen at stage $t+1$ as $T^{t+1}_{e+1}(\sigma * i)$, contrary to the fact that the e-state s_A was final.

Since the construction is recursive, s can be found recursively. Moreover, since $f(t) \geq x$ for any $t \geq s$ as just proved, we have

$$x \in C \;\Leftrightarrow\; x \in C_s.$$

Thus C is recursive, which is a contradiction. $\square$

Exercise XI.4.10 *Below any nonzero r.e. degree, there are infinitely many minimal degrees.* (Shoenfield) (Hint: let C be r.e. and nonrecursive, and suppose by induction that we have $B_1, \ldots, B_n$, all recursive in C, and of distinct minimal degrees. Let A be an r.e. nonrecursive set with $A \leq_T C$ and no B_i recursive in A. Let B_{n+1} be any set recursive in A of minimal degree.)

Corollary XI.4.11 *For any nonzero r.e. degree $\boldsymbol{a}$, the theories of the r.e. degrees below $\boldsymbol{a}$ and of the degrees below $\boldsymbol{a}$ are not elementarily equivalent.*

Proof. By X.2.8, among the r.e. degrees below $\boldsymbol{a}$ there is no minimal degree. □

The corollary uses priority in the proof of both parts. The priority-free proof of its special case $\boldsymbol{a} = \boldsymbol{0'}$ XI.3.3, based on the Diamond Theorem XI.3.1, does not work in general. If $\boldsymbol{b}$ and $\boldsymbol{c}$ are a minimal pair of r.e. degrees (X.6.5), then the Diamond Theorem holds in the r.e. degrees (and in the degrees) below $\boldsymbol{a} = \boldsymbol{b} \cup \boldsymbol{c}$.

Corollary XI.4.12 *Every degree $\boldsymbol{a} < \boldsymbol{0'}$ has a minimal cover in $\mathcal{D}(\leq \boldsymbol{0'})$.*

Proof. The result follows by relativization of XI.4.9 above $\boldsymbol{a}$, since $\boldsymbol{0'}$ is r.e. in $\boldsymbol{a}$. □

Having seen in Section 2 that permitting sometimes works not only below nonrecursive r.e. degrees but also below degrees in $\overline{\mathbf{GL}}_2$, one can ask whether this is the case for minimal degrees. Jockusch [1981] has shown that it is not so, since there is a degree in $\overline{\mathbf{GL}}_2$ which does not bound minimal degrees. Lerman [1986] has proved that $\mathbf{GH}_1$ is the only class of the Generalized Jump Hierarchy all of whose members bound minimal degrees. Moreover, the same result also holds below $\mathbf{0}'$, and thus *not every degree in $\mathbf{H}_2$ bounds a minimal degree*.

Exercise XI.4.13 *Every degree in $\mathbf{GH}_1$ bounds a minimal degree.* (Cooper [1973], Jockusch [1977]) (Hint: this does not require the full approximation method, but only a modification of XI.4.5. If the degree of B is in $\mathbf{GH}_1$, then the recursive sets are uniformly recursive in B by XI.1.4.c, so the diagonalization can be carried out recursively in it. To see if σ_s has a proper extension on T_i^s is recursive in $\mathcal{K}$, hence it can be recursively approximated. The problem is that when we look at approximations of it, we could well think that σ_s has extensions on T_i^s for every s, even if this fails in reality for every s. But suppose, by the Fixed-Point Theorem, that we already have A, and look at the question: 'does some initial segment of A have extensions on T_i^s?' This is a one-quantifier question on $A \oplus \mathcal{K}$, hence it is recursive in $(A \oplus \mathcal{K})'$. If $A \leq_T B$ by construction, the question is then recursive in $(B \oplus \mathcal{K})' \equiv_T B'$, and it can be approximated recursively in B. Thus, at stage $s+1$, what we really do is to search for either an extension of σ_s on T_i^s or a later stage in which the approximation tells us that no such extension exists.)

For more information on minimal degrees below $\mathbf{0}'$ see Epstein [1975] and Posner [1981].

The initial segments of the degrees below $\mathbf{0}'$ $\star$

The natural step after embedding minimal degrees below $\mathbf{0}'$ is to look for more complicated initial segments. The first result was obtained by Epstein [1981], who showed that ω is embeddable. The final result is that *the ordinals isomorphic to initial segments below $\mathbf{0}'$ are exactly the ordinals $\alpha \leq \omega_1^{ck}$*. The condition is easily seen to be necessary, since if $\boldsymbol{a} \leq \mathbf{0}'$ and $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ is isomorphic to a successor ordinal, then this is an arithmetical ordinal and hence (by a theorem of Spector, see Volume III) a recursive ordinal. Thus the greatest ordinal that can be isomorphic to an initial segment below $\mathbf{0}'$ is ω_1^{ck}. Conversely, Lerman [1983] has proved that every recursive topped linear ordering is embeddable as an initial segment below $\mathbf{0}'$. Since there is one such ordering with a well-ordered initial segment of length ω_1^{ck} (see Volume III), it follows that there is an initial segment below $\mathbf{0}'$ with ω_1^{ck} as its order type

A characterization of general initial segments below $\mathbf{0}'$ might be difficult to obtain for the following reasons. From XI.3.16, it follows that every topped lattice embeddable as an initial segment below $\mathbf{0}'$ must be $\emptyset^{(3)}$-presentable, and every topped uppersemilattice must be $\emptyset^{(4)}$-presentable. Since there are lattices which are $\emptyset^{(4)}$-presentable but not $\emptyset^{(3)}$-presentable, *not every $\emptyset^{(4)}$-presentable uppersemilattice is embeddable as an initial segment below $\mathbf{0}'$*. On the other hand, *there are uppersemilattices $\emptyset^{(4)}$-presentable but not $\emptyset^{(3)}$-presentable which are embeddable* (e.g. $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ itself, see Shore [1981a]). Thus the characterization of initial segments embeddable below $\mathbf{0}'$ is not trivial.

On the positive side, Lerman [1983] has proved that *every $\emptyset^{(2)}$-presentable (in particular, every finite) uppersemilattice is embeddable as an initial segment below $\mathbf{0}'$*.

XI.5 Global Properties

In this section we look at global properties of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$, following the path set up in Section V.7.

Definability from parameters

By analyzing the proof of V.7.1, we obtain the following effective version of it.

Proposition XI.5.1 (Slaman and Woodin [1986], Odifreddi and Shore [1991]) *Every uniformly low antichain is definable in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ from finitely many parameters, in a uniform way.*

Proof. Given an antichain $\mathcal{C} = \{\boldsymbol{c_n}\}_{n\in\omega}$ of degrees below $\mathbf{0}'$, we want to find parameters below $\mathbf{0}'$ coding it. We will analyze the proof of V.7.1 and discover conditions on $\mathcal{C}$ that will make the same proof work. We will only indicate the required changes.

Recall that $\mathcal{C}$ is going to be defined from three parameters A, B and C (of degrees $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$), of which $C = \oplus_{n\in\omega} C_n$ for $C_n \in \boldsymbol{c_n}$. We define A and B by finite extensions σ_s and τ_s, defined on the same elements. To make A and B recursive in $\emptyset'$, the construction will have to define σ_s and τ_s recursively in $\emptyset'$. Of course, C too will have to be recursive in $\emptyset'$. This imposes a first restriction on $\mathcal{C}$, which however is not strong enough yet for the rest of the proof to work.

Recall that we are trying to define $\mathcal{C}$ as

$$\boldsymbol{x} \in \mathcal{C} \ \Leftrightarrow \ \boldsymbol{x} \leq \boldsymbol{c} \wedge P(\boldsymbol{x}) \wedge \neg(\exists \boldsymbol{z})(\boldsymbol{z} < \boldsymbol{x} \wedge P(\boldsymbol{z})),$$

where

$$P(\boldsymbol{x}) \ \Leftrightarrow \ \boldsymbol{x} \neq (\boldsymbol{x} \cup \boldsymbol{a}) \cap (\boldsymbol{x} \cup \boldsymbol{b}),$$

i.e. as the set of minimal solutions of P below $\boldsymbol{c}$. We analyze the modifications imposed on the two parts of the proof separately.

1. *$\boldsymbol{c_n}$ is a solution of P*
   This is achieved by having, for every n, a set D_n such that

   $$D_n \leq_T C_n \oplus A, C_n \oplus B \ \text{ but } \ D_n \not\leq_T C_n.$$

   We let

   $$D_n(x) = \begin{cases} 0 & \text{if } x \notin C_n \\ A_n(x) & \text{otherwise,} \end{cases}$$

   so that $D_n \leq_T C_n \oplus A$ by definition, and $D_n \leq_T C_n \oplus B$ because A_n and B_n will differ only finitely on elements of C_n, by construction. It remains to ensure, by diagonalization, that $D_n \not\leq_T C_n$, i.e. $D_n \not\leq_T \{e\}^{C_n}$ for every e. We take care of this requirement at some fixed stage $s+1$, by taking an $x \in C_n$ such that σ_s and τ_s are not yet defined on $\langle n, x\rangle$, and by defining

   $$\sigma_{s+1}(\langle n,x\rangle) = \tau_{s+1}(\langle n,x\rangle) \simeq \begin{cases} 1 - \{e\}^{C_n}(x) & \text{if } \{e\}^{C_n}(x)\downarrow \\ 0 & \text{otherwise.} \end{cases}$$

   This step imposes a new restriction on $\mathcal{C}$. Since the construction has to be recursive in $\emptyset'$, we need to be able to decide recursively in $\emptyset'$ whether

$\{e\}^{C_n}(x)\downarrow$, uniformly in n. In other words, the set $\{C_n\}_{n\in\omega}$ must be uniformly low, that is $C = \oplus_{n\in\omega} C_n$ must be low, which is the restriction in the statement of the theorem. In particular, this implies that C is recursive in $\emptyset'$, as required above.

2. ***c_n* *is a minimal solution of* P**
We will work with a fixed list $\{X_m\}_{m\in\omega}$ of the sets recursive in C. For example, we can let $X_m(x)$ be $\{m\}^C(x)$ unless there is a $y \leq x$ for which $\{m\}^C(x)$ is either undefined or different from 0 or 1, in which case $X_m(x) \simeq 0$. Note that $\{X_m\}_{m\in\omega}$ is uniformly recursive in $\emptyset'$. Indeed, we can decide recursively in $\emptyset'$ whether $\{m\}^C(x)$ converges, because C is low.

Recall that we want, for each m,

$$D \leq_T X_m \oplus A, X_m \oplus B \;\wedge\; D \not\leq_T X_m \;\Rightarrow C_n \leq_T X_m, \text{ for some } n.$$

The requirements $R_{e,m}$ are

$$\{e\}^{X_m\oplus A} \simeq \{e\}^{X_m\oplus B} \simeq D \;\wedge\; D \not\leq_T X_m \;\Rightarrow\; C_n \leq_T X_m, \text{ for some } n.$$

We will briefly analyze the proof of V.7.1, keeping the same notation. At the step $s+1$ devoted to $R_{e,m}$, we looked for three possible cases:

- *we can make* $\{e\}^{X_m\oplus A}$ *not total*
In other words, there is a string σ compatible with θ_s, such that

$$(\exists y)(\forall \tau \supseteq \sigma)(\tau \text{ compatible with } \theta_s \Rightarrow \{e\}^{X_m\oplus\tau}(y)\uparrow).$$

- *otherwise, but we can force a special kind of disagreement*
In other words, there are two strings σ and τ compatible with θ_s and ϑ_s, respectively, which e-split on some x, i.e.

$$\{e\}^{X_m\oplus\sigma}(x) \text{ and } \{e\}^{X_m\oplus\tau}(x) \text{ converge and are different,}$$

and agree on elements of C_n, on the n-th column, for any $n < s$.
- *otherwise.*

A short argument shows that the second question is recursive in $\emptyset'$. The reason is that we ask for the existence of σ and τ satisfying certain recursive conditions involving X_m and $\oplus_{n<s} C_n$ (the latter are coded, up to a finite difference, in θ_s and ϑ_s). We thus have a question which is r.e. in $X_m \oplus (\oplus_{n<s} C_n)$, and which would be recursive in $\emptyset'$ if such a set were low. Now there is no problem for the C_n's, since they are uniformly

recursive in C which is low. As for X_m, by definition it is either a finite set or $\{m\}^C$ (if a total characteristic function). In the first case it is recursive and there is no problem. In the second case questions r.e. in X_m can be reduced to questions r.e. in C itself, by using the reduction coded by m. But we do not know which of the two cases applies. So, having a question r.e. in $X_m \oplus (\oplus_{n<s} C_n)$, we simultaneously look for an answer to it pretending that X_m is $\{m\}^C$, and for an input y such that $\{m\}^C(y) \not\simeq 0, 1$ (possibly because undefined). If we first find such a y, then we know that X_m is a finite set (and we also know which one), so we can rerun the process with the right set. Otherwise, we answer the question by using an initial segment of X_m which is correct (either because X_m is really $\{m\}^C$, or because it is a finite set extending such a segment, since the two sets differ after the first such y).

There is however little hope of showing directly that the question asked in the first case above is recursive in $\emptyset'$, since it uses two quantifiers. We thus have to modify the construction a bit. Recall that we needed to make $\{e\}^{X_m \oplus A}$ not total whenever possible, to be able to argue at the end that if the diagonalization of the second step did not work, and if $\{e\}^{X_m \oplus A} \simeq D \not\leq_T X_m$, then there were infinitely many pairs of strings e-splitting and differing on exactly one element of C_m (see p. I.353, case 2). What we can do now is to mix these two parts of the construction, and ask for convergent computations only whenever we need them (this requires only one quantifier), instead of looking directly for a way of making the function not total.

The construction is as follows. At the end, we will let $A = \bigcup_{s \in \omega} \sigma_s$ and $B = \bigcup_{s \in \omega} \tau_s$. We start with $\sigma_0 = \tau_0 = \emptyset$. At stage $s+1$, let σ_s and τ_s be given. We do something only in the following cases:

- $s = 3\langle e, n\rangle$
  Then we diagonalize against $\{e\}^{C_n}$ as stated above. We look for an $x \in C_n$ such that both σ_s and τ_s are not defined on $\langle n, x\rangle$, and let
$$\sigma_{s+1}(\langle n, x\rangle) = \tau_{s+1}(\langle n, x\rangle) \simeq \begin{cases} 1 - \{e\}^{C_n}(x) & \text{if } \{e\}^{C_n}(x)\downarrow \\ 0 & \text{otherwise.} \end{cases}$$

- $s = 3\langle e, m\rangle + 1$
  Then we try to vacuously satisfy $R_{e,m}$ by looking for a special disagreement, i.e. for two strings $\sigma \supseteq \sigma_s$ and $\tau \supseteq \tau_s$ agreeing on elements of C_n for every $n < s$, and such that for some x
$$\{e\}^{X_m \oplus \sigma}(x) \not\simeq \{e\}^{X_m \oplus \tau}(x),$$

with both sides convergent. We argued above that the question of whether there are such σ, τ and x is recursive in $\emptyset'$. If there are such strings, then we choose the least ones as σ_{s+1} and τ_{s+1}, respectively.

- $s = 3\langle e, m, n\rangle + 2$
  Then we try to make sure that either $\{e\}^{X_m \oplus A}$ is not total, or there are infinitely many e-splitting pairs agreeing on only one element (at this stage, i.e. for the present n, we try to ensure the existence of one).

  See if there are e-splitting extensions σ and σ' of σ_s agreeing on elements of C_n for every $n < s$, with computations relative to X_m. If not, do nothing (since at the end $\{e\}^{X_m \oplus A}$ is going to be recursive in X_m and $R_{e,m}$ is vacuously satisfied). Otherwise, choose σ, σ' and x such that

  $$\{e\}^{X_m \oplus \sigma}(x) \not\simeq \{e\}^{X_m \oplus \sigma'}(x),$$

  with both sides converging. As above, this step is recursive in $\emptyset'$. Then (as on p. I.533) interpolate between σ and σ' by a sequence of strings $\sigma^0, \ldots, \sigma^i$ each differing from the following one on just an element and such that $\sigma^0 = \sigma$ and $\sigma^i = \sigma'$. See if there are strings $\mu^0, \ldots, \mu^i$ such that, for $0 \le j \le i$,

  $$\{e\}^{X_m \oplus \sigma^j * \mu^0 * \cdots * \mu^j}(x)\downarrow .$$

  If so, then the strings in the sequence $\sigma^j * \mu^0 * \cdots * \mu^i$ $(0 \le j \le i)$ differ by exactly one element, and since the two extremes split on x, so do two successive strings. We have thus found one pair of strings as required.

  Otherwise, there is j such that

  $$(\forall \mu \supseteq \sigma_j * \mu^0 * \cdots * \mu^{j-1})(\{e\}^{X_m \oplus \mu}(x)\uparrow).$$

  We can then choose $\sigma_{s+1} = \sigma_j * \mu^0 * \cdots * \mu^{j-1}$, thus ensuring that the function $\{e\}^{X_m \oplus A}$ will not be total. To have A and B differing only finitely on each column, τ_{s+1} is extended to agree with σ_{s+1} on every element on which the latter is defined but the former is not.

With these modifications, the proof that $R_{e,m}$ is satisfied can now be carried out exactly as in V.7.1. In particular, either we make $\{e\}^{X_m \oplus A}$ and $\{e\}^{X_m \oplus B}$ different at stage $3\langle e, m\rangle$+2, or $\{e\}^{X_m \oplus A} \le_T X_m$, or at some stage $3\langle e, m, n\rangle$+3 we make $\{e\}^{X_m \oplus A}$ not total, or there are infinitely many e-splitting pairs differing on some element of C_n, and thus there is an infinite subset of C_n which is recursive in X_m, and hence $C_n \le_T X_m$ if we choose the C_n's as introreducibile sets. □

We can now produce an effective version of V.7.2.

Theorem XI.5.2 Definability from parameters (Slaman and Woodin [1986], Odifreddi and Shore [1991]) *Every countable relation such that the recursive union of (representatives of) its projections is low is definable in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ from finitely many parameters, in a uniform way.*

Proof. The proof is similar to that of V.7.2. Given a uniformly low set $\mathcal{C} = \{\boldsymbol{c_n}\}_{n\in\omega}$ of degrees, we spread it out to a uniformly low set of incomparable degrees, which we know how to define.

By hypothesis, there is a low degree $\boldsymbol{c}$ containing $C = \oplus_{n\in\omega} C_n$, with $C_n \in \boldsymbol{c_n}$. Define a set $\mathcal{A} = \{\boldsymbol{a_n}\}_{n\in\omega}$ of pairwise incomparable degrees not introducing any new relation on degrees of $\mathcal{C}$, and such that not only $\{\boldsymbol{a_n}\}_{n\in\omega}$, but also $\{\boldsymbol{c_n} \cup \boldsymbol{a_n}\}_{n\in\omega}$ are uniformly recursive in a low degree. This is possible by standard methods (combined with the searching procedure for X_m described above), because $\mathcal{C}$ is uniformly low.

We now spread out $\mathcal{C}$ by using an infinite subset $\mathcal{A}^* = \{\boldsymbol{a_{f_i(n)}}\}_{n\in\omega}$ of $\mathcal{A}$, with $f_i(n) = \langle i, n\rangle$ for any fixed i. Clearly $\mathcal{A}^*$ and $\mathcal{C}^* = \{\boldsymbol{c_n} \cup \boldsymbol{a_{f_i(n)}}\}_{n\in\omega}$ are uniformly low antichains, by the choice of $\mathcal{A}$ (because the function f_i is recursive and one-one, for any fixed i). Then both $\mathcal{A}^*$ and $\mathcal{C}^*$ are definable with parameters, by the previous proposition. It follows, as in V.7.2, that $\mathcal{C}$ and the map f_i^* on it induced by f_i are definable by

$$\begin{aligned} \boldsymbol{x} \in \mathcal{C} \quad &\Leftrightarrow \quad \boldsymbol{x} \leq \boldsymbol{c} \ \wedge \ (\exists \boldsymbol{a} \in \mathcal{A}^*)(\boldsymbol{x} \cup \boldsymbol{a} \in \mathcal{C}^*) \\ f_i^*(\boldsymbol{x}) = \boldsymbol{y} \quad &\Leftrightarrow \quad \boldsymbol{x} \in \mathcal{C} \ \wedge \ \boldsymbol{x} \cup \boldsymbol{y} \in \mathcal{C}^*. \end{aligned}$$

The rest of the proof for countable relations which are uniformly low in the sense that the recursive union of their projections is low, is as in V.7.2. □

Note that we cannot expect, as in the general case of V.7.2, to obtain all possible relations on degrees below $\mathbf{0}'$ definable from parameters below $\mathbf{0}'$, simply because there are many such relations ($2^{\aleph_0}$) but few parameters (countably many). Since everything definable over $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ from parameters is arithmetical in them, the best one could hope for would be to define all arithmetical sets of degrees below $\mathbf{0}'$ from parameters.

Despite the restrictions on the hypothesis, this result does provide some nice definability results. The following is an interesting example.

Corollary XI.5.3 *$\boldsymbol{\mathcal{R}}$ is definable in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ from parameters.*

Proof. Consider the master r.e. set

$$\mathcal{K}_0 = \{\langle x, e\rangle : x \in \mathcal{W}_e\},$$

i.e. the infinite join of all the r.e. sets. By the Sacks Splitting Theorem X.6.13, $\mathcal{K}_0$ is the disjoint union of two low r.e. sets A and B, which can be thought of as the infinite joins of two uniformly low classes of r.e. sets $\mathcal{A}$ and $\mathcal{B}$. By the theorem above, $\mathcal{A}$ and $\mathcal{B}$ are then definable with parameters below $\mathbf{0}'$. Since $\mathcal{W}_e$ is the e-th column of $\mathcal{K}_0$, and thus is the join of the e-th columns of A and B, we have that the set of r.e. degrees is defined over $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ by the formula

$$\varphi(\boldsymbol{x}) \ \Leftrightarrow \ (\exists \boldsymbol{a} \in \mathcal{A})(\exists \boldsymbol{b} \in \mathcal{B})(\boldsymbol{x} = \boldsymbol{a} \cup \boldsymbol{b}). \quad \square$$

The complexity of the theory of degrees below $\mathbf{0}'$

The local version of Simpson's Theorem (V.7.3) reads as follows.

Theorem XI.5.4 Shore's Theorem (Shore [1981a]) *The first-order theory of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ has the same degree (and actually the same isomorphism type) as the theory of First-Order Arithmetic.*

Proof. We prove that the two theories have the same m-degree by interpreting each in the other, thus providing faithful translations that will preserve theorems. Since the translations will actually be one-one, the theories will have the same 1-degree, and hence will be recursively isomorphic by III.7.13.

One direction is clear, since every formula about the ordering of degrees below $\mathbf{0}'$ can be interpreted in a natural way as a formula about sets of integers recursive in $\emptyset'$, and hence about their indices relative to $\emptyset'$. Thus the theory of degrees below $\mathbf{0}'$ is interpretable in First-Order Arithmetic.

For the converse, we want to show that First-Order Arithmetic is interpretable in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$. The proof elaborates the ideas used in V.7.3 and V.7.6, and it is devoted to showing on the one hand that we can express in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ the fact that given parameters code a standard model of arithmetic, and on the other hand that such parameters exist in the degrees below $\mathbf{0}'$.

Having done this, then a sentence φ of First-Order Arithmetic is true if and only if the sentence φ^* of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ is true, where φ^* states that for any (parameters bounded by any degree strictly below $\mathbf{0}'$ and coding a) standard model of arithmetic in the degrees below $\mathbf{0}'$, the translation of φ holds. Since φ^* can be effectively obtained from φ, we thus have an m-reduction of First-Order Arithmetic to the theory of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$.

It remains to deal with models of arithmetic:

1. *standard models*
   A *model of arithmetic* is a structure $\langle A, R, f_1, f_2, f_3, a\rangle$ such that:
   (a) A is a countable set
   (b) R is a total ordering on A with a first element a and a one-one successor given by f_1, and such that a is not the successor of any element of A, and every element different from a has an immediate predecessor

(c) f_2 and f_3 satisfy the usual recursive definitions of sum and product.

A *standard model of arithmetic* is a model in which R is a well-ordering.

A standard model of uniformly low degree (with a universe consisting of mutually incomparable degrees) can be embedded in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$, since any countable partial ordering can be so embedded. Thus there are parameters $\vec{\boldsymbol{a}}$ coding A, and $\vec{\boldsymbol{c}}$ coding R and the graphs of the functions f_1, f_2, and f_3.

Moreover, given degrees $\vec{\boldsymbol{a}}$ coding a countable set of mutually incomparable degrees (intended to be the universe A of a standard model of arithmetic in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$), we can say in a first-order statement of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ that given degrees $\vec{\boldsymbol{c}}$ code a relation and (the graphs of) three functions on the set coded by $\vec{\boldsymbol{a}}$, satisfying the requirements for a model of arithmetic. This because the definition of a model requires only finitely many first-order properties.

The whole problem is thus to express the fact that the model is standard, since the notion of well-ordering is a second-order one. This would apparently require quantification over all subsets of A, and hence over the parameters coding them. But there are uncountably many such subsets, and thus not all can be coded by parameters below $\mathbf{0}'$. To be able to handle standard models, we proceed as follows. Consider the standard part (corresponding to the set of integers) of a model. If the model is standard, then this part exhausts its universe and thus every proper initial segment of the universe is finite and has a least upper bound. Conversely, if the model is not standard, then the standard part is a proper initial segment with no least upper bound. Thus the model is standard if and only if every proper initial segment of the universe has a least upper bound. This is still a second-order sentence, but it really needs only quantification over any family of subsets, one of which is the standard part of the model. If we show that the latter is coded by parameters in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ in a certain way, then we can express the fact that the model is standard by saying that every proper initial segment of the universe coded in the same way has a least upper bound.

Since the natural numbers are represented by mutually incomparable degrees, we can decide to code subsets of the universe by exact pairs coding the ideal generated by them (the subsets themselves can then be obtained by intersecting these ideals with the definable universe of the model). The problem we face is thus reduced to finding exact pairs below $\mathbf{0}'$ for the ideal generated by the standard part of a model. In general the recovery of (the ideal generated by) the standard part requires a few quantifiers in the parameters defining the model, while we only know that exact pairs below $\mathbf{0}'$ exist (by XI.3.18) for ideals that are Σ^0_3 in a degree strictly below $\mathbf{0}'$.

2. *effective models*
   We now introduce effective models of arithmetic, that allow the recovery of the ideal generated by the standard part by a procedure Σ^0_3 in the parameters defining the model.

   Given a model of arithmetic in the degrees below a degree $\boldsymbol{d}$, we indicate by $\boldsymbol{i_n}$ the (mutually incomparable) degrees of its standard part, playing the role of the integers n. To obtain effective models of arithmetic we add two degrees $\boldsymbol{ev}$ and $\boldsymbol{od}$ that allow an inductive recovery of the degrees $\boldsymbol{i_n}$, in the following way:

   - $\boldsymbol{ev}$ gives the degree representing the next integer, when acting on the degree representing an even number. Precisely,
   $$((\textstyle\bigcup_{\boldsymbol{m\leq 2n}} \boldsymbol{i_m}) \cup \boldsymbol{ev}) \cap \boldsymbol{d} = \textstyle\bigcup_{\boldsymbol{m\leq 2n+1}} \boldsymbol{i_m}. \qquad \text{(XI.1)}$$
   - $\boldsymbol{od}$ gives the degree representing the next integer, when acting on the degree representing an odd number. Precisely,
   $$((\textstyle\bigcup_{\boldsymbol{m\leq 2n+1}} \boldsymbol{i_m}) \cup \boldsymbol{od}) \cap \boldsymbol{d} = \textstyle\bigcup_{\boldsymbol{m\leq 2n+2}} \boldsymbol{i_m}. \qquad \text{(XI.2)}$$

   The following inductive procedure then produces all the degrees in the ideal generated by $\{\boldsymbol{i_n}\}_{n\in\omega}$. Start with $\boldsymbol{i_0}$, and put in S_0 all the degrees below it. Given S_{2n} (containing all the degrees below $\boldsymbol{i_0} \cup \cdots \cup \boldsymbol{i_{2n}}$), put in S_{2n+1} all the degrees below $\boldsymbol{d}$ and $\boldsymbol{x} \cup \boldsymbol{ev}$, for any $\boldsymbol{x}$ in S_{2n}. Similarly for S_{2n+2}, using $\boldsymbol{od}$ in place of $\boldsymbol{ev}$. It is immediate to verify by induction that S_n contains exactly the degrees below $\bigcup_{\boldsymbol{m\leq n}} \boldsymbol{i_m}$, and thus $\bigcup_{n\in\omega} S_n$ is the ideal generated by the standard integers.

   Notice that it is essential that we work below $\boldsymbol{d}$ to generate the standard part of the model. If we dropped the part relative to $\boldsymbol{d}$ in XI.1 and XI.2, we would obtain the degrees $\boldsymbol{ev}$ and $\boldsymbol{od}$ themselves, at the first two steps, and everything would collapse.

   The crucial fact is now the following: given any degree $\boldsymbol{e}$ bounding $\boldsymbol{i_0}$, $\boldsymbol{ev}$, $\boldsymbol{od}$ and $\boldsymbol{d}$, the set of indices of characteristic functions of sets with degree in $\bigcup_{n\in\omega} S_n$ is Σ^0_3 in $\boldsymbol{e}$.

   Indeed, '$\{e\}^E$ is a characteristic function' is $\Pi^{0,E}_2$, '$\{i\}^E \leq_T \{e\}^E$' is $\Sigma^{0,E}_3$, and the join operation is effective, i.e. there is a recursive function g such that $\{i\}^E \oplus \{e\}^E \simeq \{g(i,e)\}^E$. Thus S_n is uniformly $\Sigma^{0,E}_3$ and hence so is $\bigcup_{n\in\omega} S_n$, since the union corresponds to a further existential quantifier.

3. *definability of standard effective models*
   Since our concern is definability, it would not help much if the added conditions XI.1 and XI.2 on $\boldsymbol{ev}$ and $\boldsymbol{od}$ were not expressible by a first-order statement of degrees. As they stand, they are infinite sequences of statements but they can be turned into a single one as follows.

   First, note that we can talk of even or odd integers in the given model of arithmetic, and of the successor of a given integer. But we cannot yet talk of standard integers, since this is precisely the goal of our current proof. We will thus impose conditions on every integer. In particular, we cannot talk directly of the l.u.b. of the set of integers smaller than a given integer of the model, since this set might not be finite (if the integer is nonstandard) and l.u.b.'s do not necessarily exist for arbitrary sets of degrees.

   What we can say (uniformly in $\boldsymbol{a}$) is that the join of all degrees representing integers smaller than the one represented by $\boldsymbol{a}$ exists in the degrees below $\boldsymbol{d}$. We can then say (again in a single sentence) that this join has the given property, i.e. that when joined with $\boldsymbol{ev}$ and capped with $\boldsymbol{d}$ we have the join of the degrees representing integers smaller than the successor of the one represented by $\boldsymbol{a}$. Similarly for the conditions on $\boldsymbol{od}$. Since the previous expressions are uniform in $\boldsymbol{a}$, we can quantify over $\boldsymbol{a}$ and obtain a sentence of degree theory that ensures the required conditions.

   We can thus say in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ that the given parameters code an effective model of arithmetic. Then the standard part of the model is Σ^0_3 in the parameters, and if the parameters are bounded below $\mathbf{0}'$, then there is an exact pair for it below $\mathbf{0}'$, by XI.3.18. By the method of part 1, one can thus express in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ the fact that the parameters bounded below $\mathbf{0}'$ code a standard model of arithmetic.

4. *existence of definable standard models*
   By usual methods, it is possible to embed an effective standard model of arithmetic in the degrees below $\mathbf{0}'$, in a uniformly low way. By XI.5.2, we then know that the model can be defined in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ from parameters. By XI.3.17, the ideal generated by the standard part of the model has exact pairs below $\mathbf{0}'$. Thus parameters defining an effective standard model of arithmetic exist below $\mathbf{0}'$. □

Corollary XI.5.5 (Epstein [1979], [1981]) *The first-order theory of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ is undecidable and not axiomatizable.*

Epstein obtained the undecidability of the theory of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ by constructivizing Simpson's proof of V.7.3, while Lerman [1983] derived the same result by embedding all finite distributive lattices as initial segments below $\mathbf{0}'$. Shore's original proof of XI.5.4 used the same coding as above, but again relied on initial segments to define (with parameters) models of arithmetic. The proof we gave (from Odifreddi and Shore [1991]) avoids initial segments entirely by relying on XI.5.2. This is a very strong advantage, since embedding initial segments more complicated than minimal degrees below $\mathbf{0}'$ is an unpleasant business, much more so than in the case of embeddings in $\boldsymbol{\mathcal{D}}$, and especially so for the nondistributive lattices whose consideration is required by the effective coding we had to use.

Turning to subclasses of sentences, V.2.10 shows that the one-quantifier theory of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ is decidable, with the same decision procedure as that for $\boldsymbol{\mathcal{D}}$. Lerman and Shore [1988] have also proved that *the two-quantifier theory of* $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ *is decidable*. The analysis they use is similar to that used for $\boldsymbol{\mathcal{D}}$, i.e. extension of embeddings on one side and initial segment results on the other (see p. I.490). But in this case, the decision procedure is different from that of $\boldsymbol{\mathcal{D}}$, since the two theories are different (e.g. in $\boldsymbol{\mathcal{D}}$ there is no greatest element). Finally, *the three-quantifier theory of* $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ *is undecidable* by the same proof used for $\boldsymbol{\mathcal{D}}$ (Schmerl, see Lerman [1983]).

The proof given above produces more than we stated.

Corollary XI.5.6 (Epstein [1979], Shore [1981a]) *If* $\boldsymbol{a}$ *is an arithmetical degree above* $\mathbf{0}'$*, then the first-order theory of* $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ *has the same degree (and actually the same isomorphism type) as the theory of First-Order Arithmetic. In particular, it has degree* $\mathbf{0}^{(\omega)}$.

Proof. Every formula about the ordering of degrees below $\boldsymbol{a}$ can be interpreted, in the natural way, as a formula about sets of integers recursive in $\boldsymbol{a}$, and hence about their indices relative to any set A of degree $\boldsymbol{a}$. If $\boldsymbol{a}$ is arithmetical, then the theory of degrees below $\boldsymbol{a}$ is interpretable in First-Order Arithmetic. $\square$

In the other direction (of reaching below $\mathbf{0}'$), Shore [1981a] has also showed how to extended his result to $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ both for $\boldsymbol{a}$ r.e. nonrecursive and for $\boldsymbol{a}$ high. Kumabe [1996] has done the same for $\boldsymbol{a}$ 1-generic below $\mathbf{0}'$ (the same result holds for any arithmetical 1-generic degree). Of course, some limitations on $\boldsymbol{a}$ are needed, since e.g. if $\boldsymbol{a}$ is minimal, then $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ is decidable.

Absolute definability $\star$

First we note that, since $\mathbf{0}'$ is definable in $\boldsymbol{\mathcal{D}}$ by XI.5.16, everything definable in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ is also definable in $\boldsymbol{\mathcal{D}}$. The converse does not hold. For example, $\mathbf{I}$ is definable in $\boldsymbol{\mathcal{D}}$ (since so is any relation definable in Second-Order Arithmetic and invariant under the double jump, by the results quoted after XI.5.17), but not in $\mathcal{D}(\leq \mathbf{0}')$ (see below).

Looking at individual degrees, $\mathbf{0}$ and $\mathbf{0}'$ are obviously definable in $\mathcal{D}(\leq \mathbf{0}')$ as the least and the greatest elements, respectively. No other degree below $\mathbf{0}'$ is known to be definable, but Cooper [1997] has proved that *not every low degree below* $\mathbf{0}'$ *is definable* (that not every degree is definable follows from the existence of a nontrivial automorphism, because every definable degree is fixed under every automorphism; that not every low degree is definable follows from the additional fact that, since $\mathbf{L}_1$ is an automorphism base by XI.5.7, a nontrivial automorphism must move a low degree).

Looking at classes of degrees, a general definability result (a local version of V.7.12) has been obtained by Shore [1988], and Nies, Shore and Slaman [1998]: *a relation on degrees below* $\mathbf{0}'$ *that is invariant under the double jump is definable in* $\mathcal{D}(\leq \mathbf{0}')$ *if and only if it is definable in First-Order Arithmetic.* From this one gets, with an additional argument for $\mathbf{H}_1$, that *all jump classes* $\mathbf{L}_n$ *and* $\mathbf{H}_n$ *are definable in* $\mathcal{D}(\leq \mathbf{0}')$*, with the only possible exception of* $\mathbf{L}_1$ (whose definability problem is still open). On the other hand, *the jump class* $\mathbf{I}$ *is not definable in* $\mathcal{D}(\leq \mathbf{0}')$ because it is not definable in First-Order Arithmetic (if it were, by adding to its definition the condition of being r.e. one would obtain a definition of the intermediate r.e. degrees, contradicting X.9.18.c).

By a different method (see p. 729) Cooper [1990] has shown that $\mathcal{R}$ *is definable in* $\mathcal{D}(\leq \mathbf{0}')$, thus obtaining the absolute version of XI.5.3.

Homogeneity $\star$

We first look at $\mathcal{D}(\leq \mathbf{0}')$ as a jump-interval. The first homogeneity question is thus whether there are other jump-intervals not elementarily equivalent to it. Building on relativized versions of the definability results quoted above, one can prove as in V.7.13 that *if* $\mathcal{D}([\boldsymbol{a}, \boldsymbol{a}'])$ *is elementarily equivalent to* $\mathcal{D}([\mathbf{0}, \mathbf{0}'])$*, then* $\boldsymbol{a}$ *is low*$_2$ (Shore [1981a]). In particular, $\mathcal{D}([\mathbf{0}, \mathbf{0}'])$ and $\mathcal{D}([\mathbf{0}', \mathbf{0}''])$ are not elementarily equivalent.

We now look at $\mathcal{D}(\leq \mathbf{0}')$ as a principal ideal. The second homogeneity question is thus whether there are other principal ideals not elementarily equivalent to it. A positive answer is immediate, since there are trivial principal ideals (e.g. those generated by minimal degrees). The next result, analogous to that above for jump-intervals, provides a deeper answer: *if* $\mathcal{D}(\leq \boldsymbol{a})$ *is elementarily equivalent to* $\mathcal{D}(\leq \mathbf{0}')$*, then* $\boldsymbol{a}$ *is high*$_2$ (Shore [1981a]). In particular, $\mathcal{D}(\leq \mathbf{0}')$ and $\mathcal{D}(\leq \mathbf{0}'')$ are not elementarily equivalent.

The results quoted above relativize to degrees $\boldsymbol{b}$ which are definable in First-Order Arithmetic, but not in general (because, by V.7.19, there are cones of elementarily equivalent principal ideals, or jump-intervals). As for cones (see V.7.16), to obtain results that fully relativize one has to look at isomorphisms:

if $\boldsymbol{\mathcal{D}}([\boldsymbol{a}, \boldsymbol{a}'])$ is isomorphic to $\boldsymbol{\mathcal{D}}([\boldsymbol{b}, \boldsymbol{b}'])$, or $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ is isomorphic to $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{b})$, then $\boldsymbol{a}$ and $\boldsymbol{b}$ have the same double jump.

The *principal ideals $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ for $\boldsymbol{a}$ r.e. and nonrecursive* have received particular attention. We have already seen in XI.2.11 and XI.4.9 that a number of results about degrees below $\mathbf{0}'$ can be pushed below any such $\boldsymbol{a}$, and Shore [1981a] has done the same for XI.5.4. On the other hand, from the failure of homogeneity for principal ideals, it follows that the theory of $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ is not independent of $\boldsymbol{a}$. Specific elementary differences with $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ are the possible failures of:

- the Complementation Theorem (Cooper and Epstein [1987]),
- the Cupping Theorem (Downey [1987], Cooper [1989], Slaman and Steel [1989]),
- the Capping Theorem (Li and Yang [1998], Cooper),
- the Diamond Theorem (Slaman [199?a]).

Automorphisms $\star$

Before we discuss the existence of automorphisms of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$, we consider automorphism bases.

Proposition XI.5.7 (Jockusch and Posner [1981]) *For every $n \geq 1$, both $\mathbf{L}_n$ and $\mathbf{H}_n$ generate $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ and thus are automorphism bases for it.*

Proof. We obtain the result by a number of steps:

1. $\mathbf{L}_2$ *generates the degrees below* $\mathbf{0}'$
   If $\boldsymbol{a}$ is not low$_2$ then, by permitting below non-low$_2$ degrees and the coding method of V.2.26, we can build two 1-generic degrees $\boldsymbol{b}$ and $\boldsymbol{c}$ such that $\boldsymbol{a} = \boldsymbol{b} \cup \boldsymbol{c}$. Thus either a degree is low$_2$, or it is the join of two low$_1$, and hence low$_2$, degrees.

2. $\mathbf{H}_1$ *generates the degrees below* $\mathbf{0}'$
   By part 1, it is enough to show that $\mathbf{H}_1$ generates $\mathbf{L}_2$. By permitting below non-low$_2$ degrees and the method of XI.1.20, if $\boldsymbol{a}$ is not low$_2$, there is a minimal pair of degrees below $\boldsymbol{a}$, both with jump $\boldsymbol{a}'$. By relativization, if $\boldsymbol{a}$ is not low$_2$ in $\boldsymbol{b}$ (i.e. $\boldsymbol{a}'' > \boldsymbol{b}''$), there is a pair of degrees with g.l.b. $\boldsymbol{b}$, both with jump $\boldsymbol{a}'$. If $\boldsymbol{b}$ is low$_2$ (i.e. $\boldsymbol{b}'' = \mathbf{0}''$), then $\mathbf{0}'$ is not low$_2$ in $\boldsymbol{b}$ (since $\mathbf{0}''' > \mathbf{0}'' = \boldsymbol{b}''$), and thus there is a pair of degrees with g.l.b. $\boldsymbol{b}$, both with jump $\mathbf{0}''$, i.e. high. Then every low$_2$ degree is the meet of two high degrees.

3. *The 1-generic degrees below* $\mathbf{0}'$ *generate the degrees below* $\mathbf{0}'$
   If a given degree $\boldsymbol{a}$ is not low$_2$, then it is the join of two 1-generic degrees, by part 1. If it is low$_2$, then it is the meet of two non-low$_2$ degrees by part 2, and each of these degrees is then the join of two 1-generic degrees, again by part 1. Thus every degree is either the join of two 1-generic degrees, or the meet of two degrees, each of them being the join of two 1-generic degrees.

The result now follows from part 2 for $\mathbf{H}_n$ and from part 3 for $\mathbf{L}_n$, since every 1-generic degree below $\mathbf{0}'$ is low. □

Exercises XI.5.8 a) *For every* $\boldsymbol{c} \geq \mathbf{0}'$ *r.e. in and above* $\mathbf{0}'$*, the degrees below* $\mathbf{0}'$ *with jump* $\boldsymbol{c}$ *generate* $\mathcal{D}(\leq \mathbf{0}')$. (Jockusch and Posner [1981]) (Hint: it is enough to generate $\mathbf{L}_1$. By the methods of XI.1.20, there is a minimal pair of degrees below $\mathbf{0}'$ with jump $\boldsymbol{c}$. If $\boldsymbol{a}$ is low$_1$, by relativization $\boldsymbol{a}$ is the meet of two degrees with jump $\boldsymbol{c}$.)

b) *The following classes generate the degrees below* $\mathbf{0}'$*:* $\mathbf{I}$, $\mathbf{L}_{n+1} - \mathbf{L}_n$, $\mathbf{H}_{n+1} - \mathbf{H}_n$, *for every* n. (Hint: by part a), since they are all nonempty.)

Jockusch and Posner [1981] have proved the analogues of V.7.25 and V.7.26. In particular, *the minimal degrees below* $\mathbf{0}'$ *generate* $\mathcal{D}(\leq \mathbf{0}')$.

We now turn to automorphisms. Cooper [1997] has proved that there are at least countably many automorphisms of $\mathcal{D}(\leq \mathbf{0}')$. Slaman and Woodin [199?] have proved that there are at most countably many. Thus *there are exactly countably many automorphisms of* $\mathcal{D}(\leq \mathbf{0}')$.

It is not known whether every automorphism of $\mathcal{R}$ can be extended to one of $\mathcal{D}(\leq \mathbf{0}')$, or whether every automorphism of $\mathcal{D}(\leq \mathbf{0}')$ can be extended to one of $\mathcal{D}$. But at most one of the two options can hold, since not every automorphism of $\mathcal{R}$ can be extended to one of $\mathcal{D}$ (see p. 581).

In the opposite direction, it is known that ($\mathcal{R}$ and hence) $\mathcal{D}(\leq \mathbf{0}')$ *is an automorphism base for* $\mathcal{D}$ (Slaman and Woodin [199?]). Together with the definability of $\mathbf{0}'$ provided by XI.5.16, this shows that *every nontrivial automorphism of* $\mathcal{D}$ *induces a nontrivial one of* $\mathcal{D}(\leq \mathbf{0}')$.

Finally, Slaman and Woodin [199?] have proved that *the relations on degrees below* $\mathbf{0}'$ *that are definable in* $\mathcal{D}(\leq \mathbf{0}')$ *are exactly those that are both invariant under automorphisms of* $\mathcal{D}(\leq \mathbf{0}')$ *and definable in First-Order Arithmetic.*

Both the restriction on the number of automorphisms of $\mathcal{D}(\leq \mathbf{0}')$, and the sufficiency of the definability condition just quoted (the necessity being obvious), have been proved by Slaman and Woodin as corollaries of the so-called *biinterpretability with parameters of* $\mathcal{D}(\leq \mathbf{0}')$ *and the standard model of arithmetic*, i.e. the possibility of defining in $\mathcal{D}(\leq \mathbf{0}')$ from parameters not only a standard model of arithmetic (as in XI.5.4), but also the map taking a degree $\boldsymbol{x} \leq \mathbf{0}'$ into a set (of natural numbers in the standard model) of degree

$\boldsymbol{x}$ (a strong local version of V.7.10): then each automorphism is determined by the image of the parameters, and any definition in arithmetic of a relation on degrees can be translated into a definition in the degrees through the definable map (as in V.7.11).

Subclasses of degrees below $\mathbf{0}'$ ⋆

The structure $\boldsymbol{\mathcal{R}}$ of the *r.e. degrees* is the best-known substructure of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$. We have studied it in Chapter X and in this chapter we have continuously commented about its relationship with $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$. We now review those scattered observations and add some more facts.

First, we look at elementary differences. We know that they cannot appear at the one-quantifier level (by V.2.9 and X.1.9), but some of the examples we provided are at the two-quantifier level. Here is a comparative list of what we have proved:

$\boldsymbol{\mathcal{R}}$	$\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$
Density Theorem (X.6.3)	minimal degrees (XI.4.5)
Splitting Theorem (X.6.13)	minimal degrees (XI.4.5)
Non-Branching Theorem (X.6.8)	Branching Theorem (XI.2.13)
Non-Capping Theorem (X.6.24)	Capping Theorem (XI.3.9)
Non-Cupping Theorem (X.6.17)	Cupping Theorem (XI.3.10)
Non-Diamond Theorem (X.6.23)	Complementation Theorem (XI.3.13)

One question left open by the facts that the one-quantifier theories agree while the two-quantifier theories differ is whether each one-quantifier *formula with parameters in the r.e. degrees* (as opposed to each one-quantifier *sentence*) is true in the r.e. degrees if and only if it is true in the degrees below $\mathbf{0}'$. In technical terms, whether $\boldsymbol{\mathcal{R}}$ is a Σ_1-elementary substructure of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$. This has been answered negatively by Slaman [199?].

We now look at the way the r.e. degrees sit among the degrees below $\mathbf{0}'$. From an algebraic point of view, they are very sparse. On the one hand no segment contains only r.e. degrees, but there are lots of nontrivial segments containing no r.e. degree (XI.2.12). On the other hand, there are degrees below $\mathbf{0}'$ which are

- incomparable with every nontrivial r.e. degrees (XI.3.7)
- capping with every nontrivial r.e. degree (XI.4.6.b)
- cupping with every nontrivial r.e. degree (Li [199?], Cooper and Seetapun [199?])

(it is not known whether there are degrees below $\mathbf{0}'$ complementing every nontrivial r.e. degree). From a logical point of view, XI.5.3 shows that $\boldsymbol{\mathcal{R}}$ is definable in $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ from parameters, while Cooper [1990] has proved that it is absolutely definable (see p. 729).

As $\boldsymbol{\mathcal{R}}$ is such a small part of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$, an immediate thought is to try to reach further inside the latter by expanding the former. In the next exercises we consider two ways of doing this, leading to two hierarchies of degrees below $\mathbf{0}'$.

Exercises XI.5.9 A degree is called ***n*-r.e.** if it contains an n-r.e. set (see IV.1.18). Obviously, any n-r.e. degree is below $\mathbf{0}'$.

a) *For any $n \geq 1$, there is an $n+1$-r.e. degree which is not n-r.e.* (Cooper [1971], Lachlan) (Hint: for $n = 1$, define A and B r.e. such that $A \supseteq B$ and, for every x, e and i,

$$A - B \not\simeq \{e\}^{\mathcal{W}_x} \quad \text{or} \quad \mathcal{W}_x \not\simeq \{i\}^{A-B}.$$

Let $A_s \supseteq B_s$ and pick $a \notin A_s$ as a witness. Wait until a stage t comes such that $\{e\}_t^{\mathcal{W}_{x,t}}(a) \simeq 0$. If it never comes, then $A - B \not\simeq \{e\}^{\mathcal{W}_x}$. Otherwise, we try to satisfy the second part of the requirement. Note that the value of $\{e\}^{\mathcal{W}_x}(a)$ at stage t can change afterwards only if there is $u \in \mathcal{W}_x - \mathcal{W}_{x,t}$ used in the computation. For every $u \notin \mathcal{W}_{x,t}$ used in the computation we see if $\{i\}_t^{A_t - B_t}(u) \simeq 0$, otherwise one of them witnesses $\mathcal{W}_x \not\simeq \{i\}^{A-B}$. If so, let $a \in A_{t+1}$ and restrain all other numbers used to compute $\{i\}^{A_t - B_t}(u) \simeq 0$, for all such u's. If none of these u's is in $\mathcal{W}_x$, then $\{e\}^{\mathcal{W}_x}(a)$ at stage t is final and, since we put a in A but not in B, $A - B \not\simeq \{e\}^{\mathcal{W}_x}$. Otherwise, let u and t_0 be such that $u \in \mathcal{W}_{x,t_0} - \mathcal{W}_{x,t}$. We would like to have u witness $\mathcal{W}_x \not\simeq \{i\}^{A-B}$. We know that $\{i\}^{A_t - B_t}(u) \simeq 0$ at stage t. Now $u \in \mathcal{W}_{x,t_0}$. The problem is that we have put a into A, and this could have changed the computation, since before we had $a \notin A_t$. But it is enough to put a into B, so that $a \in A - B$. Then, if nothing else happens, $A - B$ looks the same now as it was at stage t.)

b) *For any $n \geq 1$, no n-r.e. degree is minimal.* (Lachlan) (Hint: it is enough to prove that if A is n-r.e. and nonrecursive, there is a an r.e. and nonrecursive set $B \leq_T A$. For $n = 2$, let $A = C - D$ with $C \supseteq D$ r.e. and let C be the range of f recursive and one-one. Let $B = f^{-1}(D)$. Then B is r.e. and nonrecursive, for otherwise A would be too, since $A = f(\overline{B})$.)

c) *There is a degree below $\mathbf{0}'$ that is not n-r.e. for any n.* (Hint: there are minimal degrees below $\mathbf{0}'$, by XI.4.5.)

If we indicate by $\boldsymbol{D_n}$ the structure of *degrees of n-r.e. sets*, then XI.5.9.a shows that

$$\boldsymbol{\mathcal{R}} = \boldsymbol{D_1} \subset \boldsymbol{D_2} \subset \boldsymbol{D_3} \subset \cdots \subset \boldsymbol{\mathcal{D}}(\leq \mathbf{0}'),$$

and XI.5.9.b shows that $\boldsymbol{D_n}$ *is not elementarily equivalent to* $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$, for $n \geq 1$. It is not known whether:

- $\boldsymbol{D_n}$ is elementarily equivalent to $\boldsymbol{D_m}$, for $m > n > 1$ (for $n = 1$ it is not, by Arslanov [1985a])
- $\boldsymbol{D_n}$ is definable in $\boldsymbol{\mathcal{D}(\leq 0')}$, for $n > 1$ (for $n = 1$ it is, by Cooper [1990])
- $\boldsymbol{D_n}$ is definable in $\boldsymbol{D_m}$, for $m > n \geq 1$
- the first-order theory of $\boldsymbol{D_n}$ is undecidable, and actually of the same degree as the theory of First-Order Arithmetic, for $n > 1$ (for $n = 1$ it is, by X.6.31).

The structure $\boldsymbol{D_2}$ of *degrees of differences of r.e. sets* (also called *d-r.e. degrees*) has received special attention, starting with Bukaraev [1981] and Ischmuchametov [1983], [1985]. Some of the results show that $\boldsymbol{D_2}$ *is not elementarily equivalent to* $\boldsymbol{\mathcal{D}(\leq 0')}$:

- the Downward Density Theorem, i.e. the nonexistence of minimal degrees (by XI.5.9.b)
- the Non-Capping Theorem (a corollary of the existence of noncappable r.e. degrees, X.6.24, together with the fact that every nonrecursive 2-r.e. degree bounds a nonrecursive r.e. degree, by the proof of XI.5.9.b).
- the Splitting Theorem (Cooper [1992]).

Other results show that $\boldsymbol{D_2}$ *is not elementarily equivalent to* $\boldsymbol{\mathcal{R}}$:

- the Cupping Theorem (Arslanov [1985a])
- the Diamond Theorem (Downey [1989])
- the Upward Non-Density Theorem, i.e. the existence of maximal degrees (Cooper, Harrington, Lachlan, Lempp and Soare [1991]).

For more results on $\boldsymbol{D_2}$ see Arslanov [1988], [1990], [1997], Arslanov, Lempp and Shore [1996], [1996a], Cooper [1991], Cooper, Lempp and Watson [1989], Cooper and Yi [199?], Ding and Qian [1996], [199?], [199?a], Jiang [1993], Kaddah [1993], LaForte [1995], Li and Yi [1999], Yi [199?], [199?a].

For a result on $\boldsymbol{D_2}$ with an application to global properties of $\boldsymbol{\mathcal{D}}$, see the next subsection (XI.5.13 in particular).

Exercises XI.5.10 (Arslanov [1979], [1982], Jockusch and Shore [1984]) A degree $\boldsymbol{a}$ is $\boldsymbol{n}$**-r.e.a.** (n-r.e. in and above) if there are degrees $\boldsymbol{0 = a_0 < a_1 \leq \cdots \leq a_n = a}$ such that, for every i, $\boldsymbol{a_{i+1}}$ is r.e. in $\boldsymbol{a_i}$. In particular, $\boldsymbol{a_1}$ is r.e. Obviously the n-r.e.a. degrees are not necessarily below $\mathbf{0}'$, since e.g. $\mathbf{0}^{(n)}$ is n-r.e.a.

a) *For any* $n \geq 1$, *if* A *is* n*-r.e., then its degree is* n*-r.e.a.* (Hint: for $n = 2$, let $A \supseteq B$ be r.e. $A - B$ is obviously r.e. in B, which is r.e., but this is not enough to claim that $A - B$ has 2-r.e.a. degree, because we do not have $B \leq_T A - B$ in general. Let thus f be a recursive approximation to $A - B$ that changes at most twice, and define

$$\begin{aligned} \langle x, s\rangle \in C &\quad\Leftrightarrow\quad f(x,s) \text{ has never changed before } s\text{, and it is not correct} \\ \langle x, s\rangle \in D &\quad\Leftrightarrow\quad f(x,s) \text{ has changed once before } s\text{, and it is not correct.} \end{aligned}$$

Then D is r.e. because if f has changed once and is not correct, it will change again. C is r.e. in D because if f has not changed before s and it is not correct it cannot change twice, otherwise it would return to the same value, and thus it will change exactly once, i.e. there is a greater stage t such that f has changed once before t and $\langle x,t\rangle \notin D$. Clearly $C \oplus D \leq_T A - B$, since the first condition is recursive and the second requires only a comparison with the real value of $A - B$. Finally, $A - B \leq_T C \oplus D$, because $x \in A - B$ if and only if $\langle x, 0\rangle \in C$.)

b) *There are 2-r.e.a. degrees below* $\mathbf{0}'$ *without n-r.e. sets for any n.* (Hint: there is a sequence $\{A_x\}_{x\in\omega}$ uniformly recursive in $\mathcal{K}$ that enumerates all sets n-r.e. for any n. It is thus enough to find B r.e. and A r.e. in B such that, for every x, e and i,

$$A \not\simeq \{e\}^{A_x} \quad \text{or} \quad A_x \not\simeq \{i\}^{A}.$$

The proof is similar to that of XI.5.9.a, using uniform approximations to A_n recursive in $\mathcal{K}$. The main difference is that in the previous proof an element a could at most go first in A and then out of it, because the approximations to $\mathcal{W}_x$ could change only once. Now instead the approximations to A_x could change finitely many times, with no fixed bound. Thus A is defined obliquely as

$$z \in A \;\Leftrightarrow\; (\exists v)[R(z,v) \wedge D_v \subseteq \overline{B}],$$

with B r.e. Since we are defining R r.e., we can decide to always use singletons as D_v. When we want to put an element a into A at stage $t+1$, we do it by choosing an element $a_0 \notin B_t$ and setting $R(a, v_0)$ true for $D_{v_0} = \{a_0\}$. If at a later stage we want to take a out of A, we just put a_0 into B. If then we want to put a into A again, we choose another element a_1 not yet in B, and so on. Thus we never take elements out of B and B is r.e.; and A is r.e. in B, hence of 2-r.e.a. degree.)

c) *For any* $n \geq 1$, *there is an* $n+1$*-r.e.a. degree which is not n-r.e.a.* (Hint: this follows from XI.5.9.a for $n = 1$ but, by b) above, not in general. But the proof of XI.5.9.a can be extended. For example, for $n = 2$ build A, B and C r.e. such that the degree of $(A - B) \cup C$, which is 3-r.e.a. by a) above, does not contain sets of 2-r.e.a. degree, i.e. for each x and y,

$$(A - B) \cup C \not\equiv_T \mathcal{W}_x^{\mathcal{W}_y} \quad \text{or} \quad \mathcal{W}_y \not\leq_T \mathcal{W}_x^{\mathcal{W}_y}.)$$

d) *For any* $n \geq 1$, *the n-r.e.a. degrees are dense.* (Hint: for $n = 2$ let $\boldsymbol{a} < \boldsymbol{b}$, with $\boldsymbol{a}$ r.e.a. $\boldsymbol{a_1}$, $\boldsymbol{b}$ r.e.a. $\boldsymbol{b_1}$, and $\boldsymbol{a_1}$ and $\boldsymbol{b_1}$ r.e. Since $\boldsymbol{a} \leq \boldsymbol{a} \cup \boldsymbol{b_1} \leq \boldsymbol{b}$, there are three cases. If $\boldsymbol{a} = \boldsymbol{a} \cup \boldsymbol{b_1}$, then both $\boldsymbol{a}$ and $\boldsymbol{b}$ are r.e.a. $\boldsymbol{a_1} \cup \boldsymbol{b_1}$: by the relativized Density Theorem X.6.3 there is $\boldsymbol{c}$ r.e.a. $\boldsymbol{a_1} \cup \boldsymbol{b_1}$ between $\boldsymbol{a}$ and $\boldsymbol{b}$, and $\boldsymbol{c}$ is 2-r.e.a. because $\boldsymbol{a_1} \cup \boldsymbol{b_1}$ is r.e. If $\boldsymbol{a} < \boldsymbol{a} \cup \boldsymbol{b_1} < \boldsymbol{b}$, it is enough to note that $\boldsymbol{a} \cup \boldsymbol{b_1}$ is r.e.a. the r.e.

degree $\boldsymbol{a_1}$, and hence 2-r.e.a. If $\boldsymbol{a} \cup \boldsymbol{b_1} = \boldsymbol{b}$, then both $\boldsymbol{a}$ and $\boldsymbol{b}$ are r.e.a. $\boldsymbol{a_1}$ since, by the relativized Density Theorem again, there is $\boldsymbol{c}$ r.e.a. $\boldsymbol{a_1}$ between $\boldsymbol{a}$ and $\boldsymbol{b}$, and $\boldsymbol{c}$ is 2-r.e.a. because $\boldsymbol{a_1}$ is r.e.)

e) *There is a degree below* $\mathbf{0}'$ *that is not n-r.e.a. for any n.* (Hint: there are minimal degrees below $\mathbf{0}'$, by XI.4.5.)

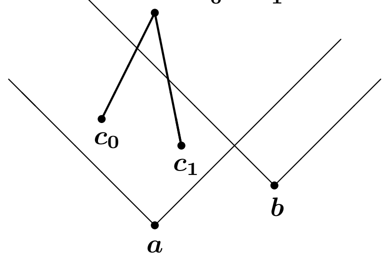
For more on the two hierarchies considered above see Carstens [1976], Arslanov, Nadyrov and Soloviev [1977], Arslanov [1979], [1982], [1989], Epstein [1979], Epstein, Haas and Kramer [1981], Selivanov [1984], [1985], Jockusch and Shore [1984], Jockusch, Lerman, Soare and Solovay [1989].

The two hierarchies can be iterated into the transfinite. However, since the Δ_2^0 sets are exactly the sets recursive in $\mathcal{K}$, appropriate relativizations of the arguments given in Chapter VII for the recursive sets (see p. 60) show that *there is no natural exhaustive hierarchy for the* Δ_2^0 *sets* (and, in general, for the Δ_n^0 sets). In particular, the transfinite iterations of the two hierarchies considered above are not natural because they heavily depend on ordinal notation.

Definability of $\mathbf{0}'$ in the degrees ⋆

In this subsection we give an account of the proof of definability of the jump operator in $\boldsymbol{\mathcal{D}}$. Although the proof of the key result is outside the scope of this book, we can give the rest of the argument. The global strategy is very similar to that used in Section XII.3 for the definability of the set $\boldsymbol{\mathcal{A}}$ of the degrees of arithmetical sets (XII.3.14), a result that is in turn easily implied by the definability of the jump (see V.8.1). We will closely follow the outline of the proof of XII.3.14, and refer to proofs of results in Section XII.3 when needed.

We first introduce, both graphically and formally, the basic notion needed in the proof.



Definition XI.5.11 *Given degrees $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$, we say that $\boldsymbol{c}$ is* **splittable over $\boldsymbol{a}$ avoiding $\boldsymbol{b}$** *if*

1. *$\boldsymbol{a}$ and $\boldsymbol{b}$ are below $\boldsymbol{c}$*
2. *$\boldsymbol{b}$ is not below $\boldsymbol{a}$*
3. *there are degrees $\boldsymbol{c_0}$ and $\boldsymbol{c_1}$ between $\boldsymbol{a}$ and $\boldsymbol{c}$ but not above $\boldsymbol{b}$, such that $\boldsymbol{c} = \boldsymbol{c_0} \cup \boldsymbol{c_1}$.*

Proposition XI.5.12 (Sacks [1963a]) *For every degree $\boldsymbol{a}$, $\boldsymbol{a} \cup \boldsymbol{0'}$ is splittable over $\boldsymbol{a}$ avoiding any $\boldsymbol{x} \leq \boldsymbol{a} \cup \boldsymbol{0'}$ such that $\boldsymbol{x} \not\leq \boldsymbol{a}$.*

Proof. By the Sacks Splitting Theorem X.6.13, given C r.e. and B r.e. nonrecursive there are C_0 and C_1 r.e. such that $C \equiv_T C_0 \oplus C_1$ and $B \not\leq_T C_i$ $(i = 0, 1)$. The result can be strengthened to any $B \in \Delta_2^0$, in particular to any $B \leq_T C$, and thus every r.e. degree is splittable over $\mathbf{0}$, avoiding any nonrecursive degree below $\mathbf{0}'$.

By relativizing to $\boldsymbol{a}$, we obtain that any degree r.e. in $\boldsymbol{a}$ and above it is splittable over $\boldsymbol{a}$, avoiding any degree nonrecursive in $\boldsymbol{a}$ and below $\boldsymbol{a'}$. Since $\mathbf{0}'$ is r.e., $\boldsymbol{a} \cup \boldsymbol{0'}$ is r.e. in $\boldsymbol{a}$ and above it, which proves the result. $\square$

The previous result provides examples of splittable degrees, while the next one goes in the opposite direction.

Proposition XI.5.13 (Cooper [1990]) *There are two r.e. sets B and C such that $B - C$ is not splittable over D avoiding X, for some sets D and X recursive in $B - C$ and such that $X \not\leq_T D$.*

This is the main result on which the remaining part of the argument rests, and it takes the same place here as the existence of minimal degrees below $\mathbf{0}'$ in XII.3.14. Although, as we have already announced, its proof is too complicated for this book, we now show how the remaining part of the argument goes.

Definition XI.5.14 *U mapping sets to sets is an* **unsplitting operator** *if, for every A, $A \leq_T U(A)$ and there is $X \not\leq_T A$ such that $U(A)$ is above X and not splittable over A avoiding X.*

The next result is the analogue of XII.3.12.

Proposition XI.5.15 (Cooper [1990]) *There is an unsplitting operator that is a 2-hop, i.e. of the form $H_e(H_i(A))$ for some e and i.*

Proof. This is an immediate consequence of XI.5.13 and it is proved as in XII.3.12:

1. *There is an unsplitting operator U such that $U(A)$ is 2-r.e. in A*
   By relativization of XI.5.13, because differences of r.e. sets are 2-r.e.

2. *There is an unsplitting operator U that is a 2-hop*
   By XI.5.10.a, since 2-r.e. sets have 2-r.e.a. degrees, and hence are 2-hops, up to degree. $\square$

Theorem XI.5.16 Cooper's Theorem (Cooper [1990]) $\mathbf{0}'$ *is definable in* $\boldsymbol{\mathcal{D}}$*. Precisely,* $\mathbf{0}'$ *is the greatest element of*

$\boldsymbol{\mathcal{C}} = \{\boldsymbol{b} : (\forall \boldsymbol{a})(\boldsymbol{a} \cup \boldsymbol{b}$ *is splittable over* $\boldsymbol{a}$ *avoiding any* $\boldsymbol{x} \leq \boldsymbol{a} \cup \boldsymbol{b}$ *s.t.* $\boldsymbol{x} \not\leq \boldsymbol{a})\}$.

Proof. Since XI.5.12 shows that $\mathbf{0}'$ is an element of $\boldsymbol{\mathcal{C}}$, it remains to prove that $\boldsymbol{\mathcal{C}}$ contains only degrees below $\mathbf{0}'$, i.e. that if $\boldsymbol{b} \not\leq \mathbf{0}'$, then there is $\boldsymbol{a}$ such that $\boldsymbol{a} \cup \boldsymbol{b}$ is not splittable over $\boldsymbol{a}$ avoiding $\boldsymbol{x}$, for some $\boldsymbol{x} \leq \boldsymbol{a} \cup \boldsymbol{b}$ such that $\boldsymbol{x} \not\leq \boldsymbol{a}$.

The proof proceeds as in XII.3.14, only at level 2 instead of at level ω in the Hops Hierarchy. The idea is to obtain

$$\boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{u}(\boldsymbol{a})$$

where $\boldsymbol{u}(\boldsymbol{a})$ is the degree of $U(A)$ for the 2-hop U of XI.5.15. Then we know that $\boldsymbol{u}(\boldsymbol{a})$ is not splittable over $\boldsymbol{a}$ avoiding some $\boldsymbol{x} \leq \boldsymbol{u}(\boldsymbol{x})$ such that $\boldsymbol{x} \not\leq \boldsymbol{a}$.

By an argument similar to but simpler than that in XII.2.22, we can obtain a 2-Hops Inversion Theorem that gives a degree $\boldsymbol{a}$ such that

$$\boldsymbol{u}(\boldsymbol{a}) = \boldsymbol{c},$$

for any given degree $\boldsymbol{c}$ above $\mathbf{0}^{(2)}$.

Moreover, if $\boldsymbol{c}$ is also above $\boldsymbol{b}$, then from (the extension of) the Cupping Theorem (quoted after) XI.3.10 we know how to obtain a degree $\boldsymbol{a}$ such that

$$\boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{c}.$$

Thus, consider any $\boldsymbol{c} \geq \boldsymbol{b} \cup \mathbf{0}^{(2)}$, which is above both $\mathbf{0}^{(2)}$ and $\boldsymbol{b}$. We have to show that it is possible to combine the two results just quoted to produce a degree $\boldsymbol{a}$ such that, simultaneously,

$$\boldsymbol{u}(\boldsymbol{a}) = \boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{c}.$$

We are thus reduced to proving the following result: *given $B \not\leq_T \emptyset'$ and C such that $B \oplus \emptyset^{(2)} \leq_T C$, there is a set A such that*

$$U(A) \ \equiv_T \ A \oplus B \ \equiv_T C.$$

Again this can be obtained by an argument similar to but simpler than that in XII.3.14, using in particular the special representation of U as a 2-r.e. operator obtained in XI.5.15 (since the result is not known to hold in general for 2-hops). $\square$

Corollary XI.5.17 *The jump operator is definable in $\mathcal{D}$.*

Proof. By relativization of XI.5.16. □

From the corollary, it follows that the global results for $\mathcal{D}'$ quoted in Section V.8 become outright results for $\mathcal{D}$. In particular:

1. **definability**
   Every relation on degrees above $\mathbf{0}^{(3)}$ which is definable in Second-Order Arithmetic is definable in $\mathcal{D}$.

2. **homogeneity**
   If $\mathcal{D}(\geq \boldsymbol{a})$ is elementarily equivalent to $\mathcal{D}$, then $\boldsymbol{a}$ is low$_3$.

   If $\mathcal{D}(\geq \boldsymbol{a})$ is isomorphic to $\mathcal{D}(\geq \boldsymbol{b})$, then $\boldsymbol{a}$ and $\boldsymbol{b}$ have the same triple jump.

3. **automorphisms**
   Every automorphism is the identity on the cone above $\mathbf{0}^{(3)}$.

Slaman and Woodin [199?] have replaced the triple by the double jump, and in particular $\mathbf{0}^{(3)}$ by $\mathbf{0}^{(2)}$, using different methods (see Nies, Shore and Slaman [1998] for alternative proofs).

The proof of XI.5.12 shows that every r.e. degree belongs to $\mathcal{C}$, while XI.5.13 shows that there is a degree below $\mathbf{0}'$ (necessarily non-r.e.) that is not in $\mathcal{C}$. Cooper [1990] has improved on XI.5.13 by showing that every degree below $\mathbf{0}'$ that is not r.e. is not in $\mathcal{C}$, thus proving that *$\mathcal{R}$ is definable in $\mathcal{D}(\leq \mathbf{0}')$*. From the definability of $\mathbf{0}'$, it follows that *$\mathcal{R}$ is definable in $\mathcal{D}$*.

By relativization, the relation 'being r.e. in and above' is definable in both $\mathcal{D}(\leq \mathbf{0}')$ and $\mathcal{D}$. Cooper [1993], [1994] has improved on this by showing that *the relation 'being r.e. in' is definable in both $\mathcal{D}(\leq \mathbf{0}')$ and $\mathcal{D}$.*

XI.6 Many-One Degrees ⋆

We now briefly look at the Δ_2^0 sets from the point of view of m-reducibility. In particular, we introduce a new reducibility that allows us to define an appropriate version of the jump operator for the m-degrees, w.r.t. which the Δ_2^0 m-degrees are closed. This proves that there is no greatest Δ_2^0 m-degree, and provides an elementary difference with $\mathcal{R}_m$.

Partial many-one reducibility

The following is a useful generalization of m-reducibility.

Definition XI.6.1 (Ershov [1970]) *A is* ***pm*-reducibile** *to B ($A \leq_{pm} B$) if, for some partial recursive function φ, the following conditions are satisfied:*

1. *the domain of φ includes A, i.e. $\varphi(x)\downarrow$ when $x \in A$*
2. *φ m-reduces A to B on its domain, i.e. whenever $\varphi(x)\downarrow$ then*

$$x \in A \;\Leftrightarrow\; \varphi(x) \in B.$$

A is ***pm*-equivalent** *to B ($A \equiv_{pm} B$) if $A \leq_{pm} B$ and $B \leq_{pm} A$.*

Note that $A \leq_{pm} B$ is defined in the same way as $A \cong B$ (A is recursively equivalent to B, see Section III.7), except using many-one partial recursive functions, as opposed to one-one.

Exercises XI.6.2 a) *If A is r.e., then $A \leq_{pm} B$ for any set $B \neq \emptyset$.*
b) *If $A \leq_{pm} B$ and B is r.e., then so is A.*
c) *If $A \leq_{pm} B$ and B is Δ^0_2, then so is A.*

Clearly *pm*-reducibility extends *m*-reducibility, by introducing possible partial reductions. The exact relationship between the two notions is the following.

Proposition XI.6.3 (Ershov [1970]) *For any set A,*

$$A \leq_m B \;\Leftrightarrow\; A \leq_{pm} B \,\wedge\, \overline{A} \leq_{pm} \overline{B}.$$

Proof. If $A \leq_m B$, then $A \leq_{pm} B$ and $\overline{A} \leq_m \overline{B}$, and hence $\overline{A} \leq_{pm} \overline{B}$.

Conversely, let φ_0 and φ_1 *pm*-reduce, respectively, A to B and $\overline{A}$ to $\overline{B}$. In particular, $dom\,\varphi_0 \supseteq A$ and $dom\,\varphi_1 \supseteq \overline{A}$. Let

$$f(x) = \begin{cases} \varphi_0(x) & \text{if } \varphi_0(x)\downarrow \text{ first} \\ \varphi_1(x) & \text{if } \varphi_1(x)\downarrow \text{ first.} \end{cases}$$

Then f is total because $dom\,\varphi_0 \cup dom\,\varphi_1 \supseteq A \cup \overline{A} = \omega$. If $x \in dom\,\varphi_0$, then

$$x \in A \;\Leftrightarrow\; \varphi_0(x) \in B.$$

If $x \in dom\,\varphi_1$, then

$$x \in A \;\Leftrightarrow\; \varphi_1(x) \in B.$$

Thus $A \leq_m B$ via f. $\square$

Exercises XI.6.4 a) *$\leq_m$ implies $\leq_{pm}$.*
b) *$\leq_{btt}$ does not imply $\leq_{pm}$.* (Hint: $A \leq_{btt} \overline{A}$ always holds, but $\overline{\mathcal{K}} \not\leq_{pm} \mathcal{K}$.)
c) *$\leq_{pm}$ does not imply $\leq_T$ on the nonempty r.e. sets.* (Hint: let A and B be T-incomparable r.e. sets.)
d) *$\leq_{pm}$ coincides with $\leq_m$ on the Π^0_1 sets different from ω.* (Hint: the condition $\overline{A} \leq_{pm} \overline{B}$ is satisfied if $\overline{A}$ and $\overline{B}$ are nonempty r.e. sets.)

Exercises XI.6.5 Partial many-one degrees. The equivalence classes of sets w.r.t. pm-equivalence are called ***pm*-degrees**.

a) $\{\emptyset\}$ *is a pm-degree.*

b) *The nonempty r.e. sets constitute a single pm-degree.*

c) *The pm-degrees of* Π_1^0 *sets coincide with their m-degrees, and their structure is isomorphic to* $\boldsymbol{\mathcal{R}_m}$.

d) *The structure of pm-degrees of* Σ_2^{-1} *sets, i.e. sets which are differences of r.e. sets, is isomorphic to* $\boldsymbol{\mathcal{R}_m}$. (Hint: if $A \supseteq B$ are r.e., let A be the range of a recursive function f and let $C = f^{-1}(A - B)$. Then $A - B \equiv_{pm} C$. Moreover, $C \in \Pi_1^0$ because $C = \overline{f^{-1}(B)}$.)

Every pm-degree consists of m-degrees, because $\leq_m$ is stronger than $\leq_{pm}$. We now define, as in VI.6.1 and VI.6.5, the notion of a pm-cylinder that will allow us to isolate the greatest m-degree inside a given pm-degree.

Definition XI.6.6 (Ershov [1970]) *The pm-cylindrification of a set A is the set*

$$A^{pm} = \{\langle e, x\rangle : \varphi_e(x)\downarrow \wedge\, \varphi_e(x) \in A\}.$$

A is a ***pm*-cylinder** *if* $A \equiv A^{pm}$, *i.e.* A *and* A^{pm} *are recursively isomorphic.*

Proposition XI.6.7 (Ershov [1970]) $A \equiv_{pm} A^{pm}$. *Moreover,*

$$A \leq_{pm} B \;\Leftrightarrow\; A^{pm} \leq_1 B^{pm}.$$

Proof. $A \leq_1 A^{pm}$ via $f(x) = \langle e, x\rangle$, with e index of the identity function, and $A^{pm} \leq_{pm} A$ via $\varphi(x) \simeq \varphi_{(x)_1}((x)_2)$.

If $A^{pm} \leq_1 B^{pm}$, then $A \leq_{pm} B$, since $A \leq_1 A^{pm}$ and $B^{pm} \leq_{pm} B$. Conversely, let $A \leq_{pm} B$. Since $A^{pm} \leq_{pm} A$ and $B \leq_1 B^{pm}$, $A^{pm} \leq_{pm} B^{pm}$ via some φ, i.e. $dom\,\varphi \supseteq A^{pm}$ and, for $x \in dom\,\varphi$,

$$\begin{aligned} x \in A^{pm} &\Leftrightarrow \varphi(x) \in B^{pm} \\ &\Leftrightarrow \varphi_{(\varphi(x))_1}((\varphi(x))_2)\downarrow \wedge\, \varphi_{(\varphi(x))_1}((\varphi(x))_2) \in B. \end{aligned}$$

Let e be such that

$$\varphi_e(x) \simeq \begin{cases} \varphi_{(\varphi(x))_1}((\varphi(x))_2) & \text{if } \varphi(x)\downarrow \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Then $A^{pm} \leq_1 B^{pm}$ via $\langle e, x\rangle$, since:

- if $x \in dom\,\varphi$, then $\varphi_e(x) \simeq \varphi_{(\varphi(x))_1}((\varphi(x))_2)$ and

$$\begin{aligned} x \in A^{pm} &\Leftrightarrow \varphi_e(x)\downarrow \wedge\, \varphi_e(x) \in B \\ &\Leftrightarrow \langle e, x\rangle \in B^{pm} \end{aligned}$$

- if $x \notin dom\,\varphi$, then $\varphi_e(x)\uparrow$, and $\langle e, x\rangle \notin B^{pm}$. $\Box$

Proposition XI.6.8 (Ershov [1970]) *The following conditions are equivalent:*

1. *A is a pm-cylinder*
2. *for every set B, $B \leq_{pm} A \Rightarrow B \leq_1 A$.*

Proof. 1 implies 2 because if $B \leq_{pm} A$, then $B \leq_{pm} A^{pm}$. As in the proof of XI.6.7, $B \leq_1 A^{pm}$. If A is a *pm*-cylinder, then $A^{pm} \leq_1 A$ and hence $B \leq_1 A$.

2 implies 1 because $A \leq_1 A^{pm}$ and $A^{pm} \leq_{pm} A$ always hold, and by 2 the latter implies $A^{pm} \leq_1 A$. Thus $A \equiv_1 A^{pm}$. By III.7.13, $A \equiv A^{pm}$. $\Box$

In particular, *a pm-cylinder is a cylinder*. Also, *every pm-degree contains a greatest m-degree*, which is irreducible and consists exactly of the *pm*-cylinders in the given *pm*-degree.

The mini-jump operator

The notion of a *pm*-cylinder allows us to define a refined version of the jump operator for m-degrees, that always goes up in the m-degrees, but not as much as the Turing jump.

The main idea behind this is to note that $\leq_{pm}$ is a weak analogue of relative recursive enumerability. For example,

$$A \leq_{pm} \omega \;\Leftrightarrow\; A \text{ is r.e.}$$

because $A \leq_{pm} \omega$ means that there exists a partial recursive function φ such that $dom\,\varphi \supseteq A$ and

$$\varphi(x)\downarrow \Rightarrow\; (x \in A \Leftrightarrow \varphi(x) \in \omega)$$

But since $\varphi(x) \in \omega$ always holds when $\varphi(x)$ is defined, then $dom\,\varphi = A$ and A is r.e.

Thus one considers the sets $A \leq_{pm} B \oplus \overline{B}$ as the analogue, for m-reducibility, of the sets r.e. in B. The presence of $\overline{B}$ is due to the fact that $(p)m$-reducibility is not symmetric w.r.t. complements. In relative computations, we are instead given a complete knowledge of the oracle set.

From this point of view, *pm*-cylindrification plays the role of the jump operator. For example, $\omega^{pm} = \mathcal{K}_0$ (see p. I.150), since

$$\langle e, x\rangle \in \omega^{pm} \;\Leftrightarrow\; \varphi_e(x)\downarrow \Leftrightarrow\; \langle e, x\rangle \in \mathcal{K}_0.$$

We thus arrive at the following definition.

Definition XI.6.9 (Ershov [1970]) *The* **mini-jump** *of a set A is*

$$mj(A) = (A \oplus \overline{A})^{pm}.$$

The following properties are immediate:

1. *$A \leq_{pm} B \oplus \overline{B}$ if and only if $A \leq_1 mj(B)$*
   If $A \leq_{pm} B \oplus \overline{B}$, then $A \leq_1 A^{pm} \leq_1 (B \oplus \overline{B})^{pm}$. The converse follows from $A \leq_1 (B \oplus \overline{B})^{pm} \leq_{pm} B \oplus \overline{B}$.

2. *$A, \overline{A} \leq_1 mj(A)$*
   This is just a particular case of 1.

3. *$mj(A) \equiv mj(\overline{A})$*
   By symmetry, since $A \oplus \overline{A} \equiv \overline{A} \oplus A$.

The next result shows that the intuition that led to the definition of the mini-jump was correct.

Proposition XI.6.10 (Ershov [1970]) *For any set A, $mj(A) \not\leq_m A$ and thus $A <_m mj(A)$.*

Proof. Suppose $mj(A) \leq_m A$. Then $A \equiv_m mj(A)$, and A is a pm-cylinder. Moreover $A \equiv_m \overline{A}$, because by symmetry $\overline{A} \equiv_m mj(\overline{A})$ and $mj(A) \equiv mj(\overline{A})$.

We show that it is impossible for any pm-cylinder B^{pm} (and thus for A) to be m-equivalent to its complement. Suppose there is a recursive function f such that

$$\langle e, x\rangle \in B^{pm} \;\Leftrightarrow\; f(\langle e, x\rangle) \notin B^{pm}.$$

Let g be a recursive function such that

$$\varphi_{g(e)}(x) \simeq \varphi_{(f(\langle e,x\rangle))_1}((f(\langle e, x\rangle))_2).$$

Then

$$\langle e, x\rangle \in B^{pm} \Leftrightarrow \langle g(e), x\rangle \notin B^{pm}$$

by definition of B^{pm}. Let e be a fixed-point of g, i.e. $\varphi_e = \varphi_{g(e)}$. Then

$$\langle e, x\rangle \in B^{pm} \Leftrightarrow \langle e, x\rangle \notin B^{pm},$$

which is a contradiction. $\Box$

It is immediate to note that the mini-jump operator is invariant under m-equivalence, and thus it induces an operator on m-degrees.

Definition XI.6.11 (Ershov [1970]) *The* **mini-jump $\boldsymbol{a}'$** *of an m-degree $\boldsymbol{a}$ is the degree of $mj(A)$, for any $A \in \boldsymbol{a}$.*

The finite iterations of the mini-jump are defined by induction, as usual:

$$\boldsymbol{a}^{(0)} = \boldsymbol{a} \qquad \boldsymbol{a}^{(n+1)} = (\boldsymbol{a}^{(n)})'.$$

Exercises XI.6.12 (Ershov [1970]) a) *If $A \in \mathbf{0}_m^{(n+1)}$, then A is Σ_{n+1}^{-1}-complete.* (Hint: by induction on n. For $n = 0$, this is trivial, since Σ_1^{-1} is the set of r.e. sets. For $n = 1$, Σ_2^{-1} is the set of all differences of r.e. sets, and by IV.1.18.c, $\mathcal{K} \cdot \overline{\mathcal{K}}$ is 1-complete for Σ_2^{-1}. If $B \leq_{pm} \mathcal{K} \cdot \overline{\mathcal{K}}$, then B is a difference of r.e. sets, hence $B \leq_1 \mathcal{K} \cdot \overline{\mathcal{K}}$ by 1- completeness and $\mathcal{K} \cdot \overline{\mathcal{K}}$ is a pm-cylinder by XI.6.8. Moreover, $\mathcal{K} \cdot \overline{\mathcal{K}} \equiv_{pm} \mathcal{K} \oplus \overline{\mathcal{K}}$, and thus $\mathcal{K} \cdot \overline{\mathcal{K}}$ is the mini-jump of $\mathcal{K}$.)

b) *A set A is in the Boolean Hierarchy if and only if its m-degree is bounded by some $\mathbf{0}_m^{(n)}$.*

V.1.9 and XI.6.12 can be extended by iterating, respectively, the Turing jump and the mini-jump through the recursive ordinals, thus providing exhaustive hierarchies for Δ_1^1 and Δ_2^0.

Δ_2^0 many-one degrees

We now compare the Δ_2^0 m-degrees to other structures.

Proposition XI.6.13 *The structures of Δ_2^0 Turing and m-degrees are not elementarily equivalent.*

Proof. There is a greatest Δ_2^0 Turing degree, namely $\mathbf{0}'$. But there is no greatest Δ_2^0 m-degree, since if $A \in \Delta_2^0$, then its mini-jump is Δ_2^0 and has greater m-degree. $\square$

One can ask whether the existence of a greater element is the only significant difference between the two structures, and the next exercise shows that the answer is negative.

Exercise XI.6.14 *When restricted to Δ_2^0 sets the m-degrees are distributive, but the Turing degrees are not.* (Hint: see VI.1.8 and X.6.27.)

Concerning the way the r.e. sets sit inside the Δ_2^0 sets, we know that the respective structures look quite different from the point of view of Turing degrees.

Proposition XI.6.15 *The structures of r.e. and Δ_2^0 m-degrees are not elementarily equivalent.*

Proof. There is a greatest r.e. m-degree, namely $\mathbf{0}'_m$, but there is no greatest Δ_2^0 m-degree. $\square$

However, in this case the existence of a greatest element seems to be the only real difference between the two structures, since Denisov [1978] has announced that *the m-degrees of Δ_2^0 sets are isomorphic to the r.e. degrees without $\mathbf{0}'_m$*. This provides an answer to all the global questions about the structure of the Δ_2^0 m-degrees, in particular a characterization of it up to isomorphism (see p. 534).

æ

Chapter XII

Arithmetical Sets

In this chapter we provide a microscopic analysis of the arithmetical sets. In Section 1 we introduce the fundamental tool of **forcing** in the particularly simple environment of arithmetic. Forcing will be used throughout the rest of the book, in this as well as more general settings, and in Section 2 we provide some first applications. In Section 3 we study the **Turing degrees of arithmetical sets** and prove that they are definable in $\mathcal{D}$, thus filling a gap left in the proofs of the global results of Chapter V.

XII.1 Forcing in Arithmetic

We illustrate by an example what we would like to do. We have proved in V.2.2 that there are two Turing-incomparable degrees below $\mathbf{0}'$. Or, in terms of the Arithmetical Hierarchy, that there are two Δ^0_1-incomparable Δ^0_2 sets. The straightforward way of doing this is to build A and B such that $A \not\leq_T B$ and $B \not\leq_T A$, i.e. such that $A \not\simeq \{e\}^B$ and $B \not\simeq \{e\}^A$ for each e. The basic property on which the construction is based is continuity of the recursive functionals (II.3.13), i.e. the fact that convergent computations $\{e\}^X(x)$ only use a finite part of the characteristic function of X, and are preserved under extensions of this part.

We would now like to extend this incomparability result to higher levels in the Arithmetical Hierarchy, and prove e.g. that there are Δ^0_3 sets A and B which are Δ^0_2-incomparable, i.e. each not Δ^0_2 in the other. The problem is that *continuity does not hold in general for functionals whose graph has arithmetical complexity greater than* Σ^0_1. As a simple example, consider

$$F(\alpha, x) \simeq \begin{cases} 0 & \text{if } (\forall z)(\alpha(z)\downarrow \Rightarrow \alpha(z) \simeq 0) \\ 1 & \text{otherwise.} \end{cases}$$

To show that F is not continuous, let α be such that $F(\alpha, x) \simeq 0$ and suppose there is a finite function $u \subseteq \alpha$ such that $F(u, x) \simeq 0$; for any extension β of u which is not constantly 0 we have $F(\beta, x) \not\simeq 0$, contradicting the second clause of the definition. The source of the problem is the universal quantifier in the definition of F, which is strong enough to force consideration of all of α, instead of just a finite part of it.

Failure of continuity for arithmetical nonrecursive functionals seems to be a very serious obstacle in lifting proofs from the first to higher levels of the Arithmetical Hierarchy, but in this section we develop a method called **forcing** that will allow us to do so. This is based on an idea used by Cohen [1963] for the independence of the Continuum Hypothesis, and it consists of dealing not directly with arithmetical truth (which, as we have just argued, is not reducible to finite information), but with finite approximations to it. In some special cases (**generic sets**) the approximations will coincide with truth in the limit, and by restricting ourselves to these special cases we will be able to exploit continuity as usual. Cohen actually used the forcing method in the context of Set Theory, a more complicated setting which we will discuss later (see Volume III). The adaptation of forcing to the simpler case of arithmetic is due to Feferman [1965] and it provides a gentle introduction to the method.

Definition of forcing

We start by reviewing the definition of truth in arithmetic (IV.1.1 and IV.1.2) in a slightly more general context.

Definition XII.1.1 Definition of Truth in Arithmetic (Tarski [1936]) *Let $\mathcal{L}$ be the first-order language with equality, augmented with constants $\overline{n}$ for each number n, a set constant X together with a membership symbol $\in$ (to be used only in formulas of the kind '$z \in X$'), and a relation symbol φ_R for each recursive relation R.*

Given a closed formula φ of $\mathcal{L}$, ***φ is true for A*** *($A \models \varphi$) is inductively defined as follows:*

$$\begin{array}{lcl} A \models \varphi_R(\overline{n}_1, \ldots, \overline{n}_m) & \Leftrightarrow & R(n_1, \ldots, n_m) \\ A \models \overline{n} \in X & \Leftrightarrow & n \in A \\ A \models \neg\psi & \Leftrightarrow & \textit{not } (A \models \psi) \\ A \models \varphi_0 \vee \varphi_1 & \Leftrightarrow & A \models \varphi_0 \textit{ or } A \models \varphi_1 \\ A \models \exists x \psi(x) & \Leftrightarrow & \textit{for some } n,\ A \models \psi(\overline{n}). \end{array}$$

From the definition it follows that

$$\begin{array}{lcl} A \models \varphi_0 \wedge \varphi_1 & \Leftrightarrow & A \models \varphi_0 \text{ and } A \models \varphi_1 \\ A \models \forall x \psi(x) & \Leftrightarrow & \text{for all } n,\ A \models \psi(\overline{n}). \end{array}$$

As already mentioned, the basic idea of Cohen is to try to approximate truth by using finite approximations, i.e. by using strings in place of the whole set A. The intention is to produce a situation in which if σ tells us that φ holds, then $A \models \varphi$ for all $A \supseteq \sigma$. We will almost succeed, in the sense that this will hold for any A in a comeager set. The whole idea makes sense because in XII.1.1, the truth of atomic formulas is indeed determined by finite strings.

Definition XII.1.2 Definition of Forcing in Arithmetic (Cohen [1963], Feferman [1965]) *Let $\mathcal{L}$ be the language considered in XII.1.1. Given a closed formula φ of $\mathcal{L}$,* **σ forces φ** *($\sigma \Vdash \varphi$) is inductively defined as follows:*

$$\begin{array}{lcl} \sigma \Vdash \varphi_R(\overline{n}_1, \ldots, \overline{n}_m) & \Leftrightarrow & R(n_1, \ldots, n_m) \\ \sigma \Vdash \overline{n} \in X & \Leftrightarrow & \sigma(n) = 1 \\ \sigma \Vdash \neg\psi & \Leftrightarrow & (\forall \tau \supseteq \sigma) \text{ not } (\tau \Vdash \psi) \\ \sigma \Vdash \varphi_0 \vee \varphi_1 & \Leftrightarrow & \sigma \Vdash \varphi_0 \text{ or } \sigma \Vdash \varphi_1 \\ \sigma \Vdash \exists x \psi(x) & \Leftrightarrow & \text{for some } n,\ \sigma \Vdash \psi(\overline{n}). \end{array}$$

The definition of forcing is thus parallel to the definition of truth, except for the case of negation. The reason why we do not let

$$\sigma \Vdash \neg\psi \Leftrightarrow \text{not } (\sigma \Vdash \psi)$$

as for truth, is that σ only gives an approximation to truth, and the fact that $\sigma \Vdash \psi$ does not hold could simply mean that σ does not contain enough information to decide ψ, but some larger string could. If this does not happen, i.e. $\tau \Vdash \psi$ does not hold for any extension τ of σ, then we say that $\sigma \Vdash \neg\psi$. We might thus say that *in the case of truth, negation is negation of actuality, while in the case of forcing, negation is negation of possibility.*

Exercise XII.1.3 *$\sigma \Vdash \overline{n} \notin X$ if and only if $\sigma(n) = 0$.*

Proposition XII.1.4 *The following properties hold for all formulas φ:*

1. **Monotonicity**. *If $\sigma \Vdash \varphi$ and $\tau \supseteq \sigma$, then $\tau \Vdash \varphi$.*
2. **Consistency**. *$\sigma \Vdash \varphi$ and $\sigma \Vdash \neg\varphi$ do not both hold.*
3. **Quasi-completeness**. *$(\forall \sigma)(\sigma \Vdash \neg\varphi \ \vee\ (\exists \tau \supseteq \sigma)(\tau \Vdash \varphi))$.*

Proof. Monotonicity holds by induction on φ, using for negation the fact that every extension of τ is an extension of σ.

Consistency follows from the definition of forcing for negation.

Quasi-completeness holds because if there is no $\tau \supseteq \sigma$ forcing φ, then by definition $\sigma \Vdash \neg\varphi$. $\square$

Exercises XII.1.5 Weak forcing. We say that σ **weakly forces** φ (written $\sigma \Vdash^w \varphi$) if $\sigma \Vdash \neg\neg\varphi$.

a) $\sigma \Vdash^w \varphi$ *if and only if* $(\forall \tau \supseteq \sigma)(\exists \tau_1 \supseteq \tau)(\tau_1 \Vdash \varphi)$.

b) $\sigma \Vdash \varphi_0 \wedge \varphi_1$ *if and only if* $\sigma \Vdash^w \varphi_0$ *and* $\sigma \Vdash^w \varphi_1$.

c) $\sigma \Vdash \forall x \psi(\overline{x})$ *if and only if, for all* n*,* $\sigma \Vdash^w \psi(\overline{n})$.

d) *If* $\sigma \Vdash \varphi$*, then* $\sigma \Vdash^w \varphi$*, but not conversely.* (Hint: let $\sigma(n)$ be undefined and let φ be $\overline{n} \in X \vee \overline{n} \notin X$. Then σ does not force φ, but it weakly forces it.)

e) $\sigma \Vdash \neg\varphi$ *if and only if* $\sigma \Vdash^w \neg\varphi$*.* (Hint: if $\varphi \Vdash^w \neg\varphi$ then, given $\tau \supseteq \sigma$, there is $\tau_1 \supseteq \tau$ such that $\tau_1 \Vdash \neg\varphi$. If $\tau \Vdash \varphi$, then $\tau_1 \Vdash \varphi$ by monotonicity, contradicting consistency. Thus $(\tau \Vdash \varphi)$ does not hold for any $\tau \supseteq \sigma$, and hence $\sigma \Vdash \neg\varphi$.)

There is no particular reason to privilege disjunction and existential quantification in definition XII.1.2. If we wanted, we could define forcing for conjunction and universal quantification, and regard disjunction and existential quantification as defined symbols. Or we could define forcing for all the connectives, e.g. having by definition

$$\begin{aligned} \sigma \Vdash \exists x \psi(x) \quad &\Leftrightarrow \quad \text{for some } n,\ \sigma \Vdash \psi(\overline{n}) \\ \sigma \Vdash \forall x \psi(x) \quad &\Leftrightarrow \quad \text{for all } n,\ \sigma \Vdash \psi(\overline{n}) \end{aligned}$$

making all connectives independent. Then, for example, it would not be true that

$$\sigma \Vdash \exists x \psi(x) \ \Leftrightarrow\ \sigma \Vdash \neg\forall x \neg\psi(x).$$

The choice of the definition of forcing is usually made with an eye to applications, i.e. both to simplify technical details and to make possible results such as XII.1.6 and XII.1.10. For example, XII.1.6 does not hold for weak forcing. On the other hand, XII.1.10 implies that forcing is equal to truth on a comeager set, and thus all definitions for which the same result holds will coincide on a comeager set.

Proposition XII.1.6 Definability of forcing (Cohen [1963], Feferman [1965], Hinman [1969])

1. *$\{(\sigma, \varphi) : \varphi$ is a Σ^0_n sentence and $\sigma \Vdash \varphi\}$ is Σ^0_n-complete, hence recursive in $\emptyset^{(n)}$. Similarly for Π^0_n.*

2. *$\{(\sigma, \varphi) : \varphi$ is an arithmetical sentence and $\sigma \Vdash \varphi\}$ is recursive in $\emptyset^{(\omega)}$, but not arithmetical.*

3. *For a fixed arithmetical sentence φ, $\{\sigma : \sigma \Vdash \varphi\}$ is arithmetical.*

Proof. Part 1 is proved by induction on φ. First note that to check whether φ is an arithmetical sentence, and in this case to see if it is Σ^0_n or Π^0_n for a given n, is a recursive procedure. So is to decide whether $\sigma \Vdash \varphi$ for φ an atomic formula. The crucial cases are thus negation and existential quantification, which introduce a universal and an existential quantifier, respectively. This proves that forcing for Σ^0_n sentences is Σ^0_n. To prove completeness, consider the Σ^0_n formula

$$\varphi(x) \Leftrightarrow x \in \emptyset^{(n)}.$$

Since it does not contain the set variable X, $\varphi(\overline{x})$ is either forced by all strings (if true) or by none (if false). Then

$$x \in \emptyset^{(n)} \Leftrightarrow \emptyset \Vdash \varphi(\overline{x}).$$

(Notice that $\emptyset$ ambiguously denotes the empty set on the l.h.s., and the empty string on the r.h.s.) Since $\emptyset^{(n)}$ is a Σ^0_n-complete set, this shows that so is forcing for Σ^0_n sentences.

Given an arithmetical formula φ, we cannot use the fact that φ is logically equivalent to a Σ^0_n formula for some n, since forcing is not preserved under logical equivalence (see XII.1.5.d). However, by induction on the construction of φ we can reduce forcing for it to $\emptyset^{(n)}$, for an n depending uniformly on φ. This proves both part 3 and that forcing for arithmetical sentences is recursive in $\emptyset^{(\omega)}$. It is not arithmetical because otherwise it would be Σ^0_n for some n, and hence forcing for Σ^0_{n+1} sentences would be Σ^0_n, contradicting the proof of part 1. $\Box$

The complexity of forcing thus turns out to be similar to that of truth (see IV.1.12).

Generic sets

Having defined forcing for finite strings, we now consider sets again.

Definition XII.1.7 ***A* forces $\boldsymbol{\varphi}$ ($\boldsymbol{A \Vdash \varphi}$)** *if, for some $\sigma \subseteq A$, $\sigma \Vdash \varphi$.*

It is immediate that the global analogue of consistency holds. Indeed, if $A \Vdash \varphi$ and $A \Vdash \neg\varphi$, by monotonicity there would be a string contained in A forcing both φ and $\neg\varphi$, contradicting consistency for strings. The global analogue of quasi-completeness is the crucial property and is given by the next definition.

Definition XII.1.8 (Feferman [1965], Hinman [1969])

1. *A is* $\boldsymbol{n}$**-generic** *if, for every Σ^0_n sentence φ, $A \Vdash \varphi$ or $A \Vdash \neg\varphi$.*
2. *A is* $\boldsymbol{\omega}$**-generic** *if, for every arithmetical sentence φ, $A \Vdash \varphi$ or $A \Vdash \neg\varphi$.*

We will see in XII.1.14 that the notion of 1-genericity used in Chapter XI is equivalent to the one just introduced (for $n = 1$).

Exercises XII.1.9 a) *Every set is 0-generic.*

b) *If A is n-generic and φ is Π^0_n, then $A \Vdash \varphi$ or $A \Vdash \neg\varphi$.*

c) *A is n-generic if and only if $\overline{A}$ is.* (Hint: consider the isomorphism of sentences defined by sending '$z \in X$' into '$z \notin X$' and conversely.)

d) *If A is n-generic, every finite modification of it is n-generic.*

The next theorem is the central result of the theory.

Theorem XII.1.10 Forcing equals truth for generic sets (Cohen [1963], Feferman [1965], Hinman [1969])

1. *A is n-generic if and only if, for every Σ^0_n or Π^0_n sentence φ,*
$$A \models \varphi \;\Leftrightarrow\; A \Vdash \varphi.$$

2. *Similarly if A is ω-generic and φ is arithmetical.*

Proof. The left-to-right directions are proved by induction on φ. Negation is the only case in which forcing is not defined as truth, so we need only to deal with this. Then

$$\begin{array}{lll} & A \models \neg\psi & \\ \Leftrightarrow & \text{not } (A \models \psi) & \text{(by definition of truth)} \\ \Leftrightarrow & \text{not } (A \Vdash \psi) & \text{(by induction hypothesis)} \\ \Leftrightarrow & A \Vdash \neg\psi & \text{(by genericity of } A\text{).} \end{array}$$

To prove the converse, let e.g. φ be a Σ^0_n sentence. Since $A \models \varphi$ or $A \models \neg\varphi$ by definition of truth, by hypothesis $A \Vdash \varphi$ or $A \Vdash \varphi$, and so A is n-generic. $\Box$

Because of the way genericity is defined (every arithmetical property is decided by a finite amount of information about the generic set) all generic sets look alike, in the sense that they have the same arithmetical properties which are invariant under finite changes.

Theorem XII.1.11 Generic Existence Theorem (Cohen [1963], Feferman [1965], Hinman [1969])

1. *The family of ω-generic sets is comeager in $\mathcal{P}(\omega)$ (and in the subspace of $\mathcal{P}(\omega)$ determined by any given string).*

2. *If $n \geq 1$, then an n-generic set is neither Σ^0_n nor Π^0_n, but there are n-generic sets recursive in $\emptyset^{(n)}$.*

3. *An ω-generic set is not arithmetical, but there are ω-generic sets recursive in $\emptyset^{(\omega)}$.*

Proof. Given a string σ, we build an ω-generic set $A \supseteq \sigma$ by the finite extension method. As usual (V.3.13), this implies that the set of ω-generic sets is comeager in the subspace of $\mathcal{P}(\omega)$ determined by σ. Let $\{\psi_n\}_{n\in\omega}$ be a recursive enumeration of the arithmetical sentences. Let $\sigma_0 = \sigma$ and, given σ_s, let σ_{s+1} be a string $\tau \supseteq \sigma_s$ such that

$$\tau \Vdash \psi_s \text{ or } \tau \Vdash \neg\psi_s.$$

Note that τ exists by quasi-completeness and it can be found recursively in $\emptyset^{(\omega)}$ by XII.1.6.2. Since sentences of the form $\overline{n} \in X$ and their negations can be forced only by strings defined on n, $\bigcup_{s\in\omega} \sigma_s$ defines a set A that, by construction, is ω-generic and recursive in $\emptyset^{(\omega)}$.

Similarly, an n-generic set can be built recursively in $\emptyset^{(n)}$ as follows. Start from a recursive enumeration of the Σ^0_n sentences and notice that $\tau \Vdash \psi_s$ and $\tau \Vdash \neg\psi_s$ are Σ^0_n and Π^0_n, respectively, and hence both recursive in $\emptyset^{(n)}$, when ψ_s is Σ^0_n (by XII.1.6.1).

We now prove that if $n \geq 1$, and A is n-generic, then A is infinite but it does not have infinite Σ^0_n subsets. In particular, A is not Σ^0_n itself and an ω-generic set is not arithmetical.

- *A is infinite*
  Suppose A is finite. Then, for some m, A satisfies the following sentence φ:
  $$(\forall x \geq \overline{m})(x \notin X).$$
  Since φ is Π^0_1 and A is n-generic with $n \geq 1$, $\sigma \Vdash \varphi$ for some $\sigma \subseteq A$. Choose $\tau \supseteq \sigma$ such that $\tau(x) = 1$, for some $x \geq m$ for which σ is undefined. By the first part of the proof, there is an n-generic set $B \supseteq \tau$. But $B \supseteq \tau \supseteq \sigma$, so on the one hand B satisfies φ (by n-genericity) and should not have any elements greater than m, and on the other hand B extends τ and x is in B, which is a contradiction.

- *A has no infinite Σ^0_n subsets*
  Suppose C is an infinite Σ^0_n subset of A. Then A satisfies the following sentence φ:
  $$(\forall x)(x \in C \to x \in X).$$
  Since φ is Π^0_n and A is n-generic, $\sigma \Vdash \varphi$ for some $\sigma \subseteq A$. Choose $\tau \supseteq \sigma$ such that $\tau(x) = 0$, for some $x \in C$ for which σ is undefined (which exists because C is infinite). By the first part of the proof, there is an n-generic set $B \supseteq \tau$. But $B \supseteq \tau \supseteq \sigma$, so on the one hand B satisfies φ

(by n-genericity) and should contain C, and on the other hand B extends τ and x is not in B, which is a contradiction. $\Box$

The reader will have noticed that the proof of the negative part of the above result consists of two arguments of the same kind. They are very typical of the reasoning for generic sets.

Exercise XII.1.12 *If A is n-generic for $n \geq 1$, then it does not have coinfinite Π^0_n supsersets.*

Genericity without forcing $\star$

It is possible to give alternative, forcing-free characterizations of genericity.

Proposition XII.1.13 (Posner [1977], Jockusch [1981])

1. *A is n-generic if and only if, for any Σ^0_n set of strings S, there is $\sigma \subseteq A$ such that*
$$\sigma \in S \text{ or } (\forall \tau \supseteq \sigma)(\tau \notin S).$$
2. *Similarly if A is ω-generic and S is arithmetical.*

Proof. We only prove part 1, part 2 being similar. Let A be n-generic and $S \in \Sigma^0_n$. Consider the following φ:

$$(\exists \sigma \subseteq X)(\sigma \in S).$$

Then $\varphi \in \Sigma^0_n$ and by n-genericity $A \Vdash \varphi$ or $A \Vdash \neg\varphi$. In the first case, $A \models \varphi$ and so there is $\sigma \subseteq A$ such that $\sigma \in S$. In the second case, $\sigma \Vdash \neg\varphi$ for some $\sigma \subseteq A$. Suppose $\tau \in S$ for some $\tau \supseteq \sigma$ and let $B \supseteq \tau$ be n-generic. Then $B \Vdash \neg\varphi$ because $B \supseteq \sigma$ and so $B \models \neg\varphi$. But also $B \models \varphi$, since $\tau \subseteq B$ and $\tau \in S$, which is a contradiction.

For the converse, let $\psi \in \Sigma^0_n$ and $S = \{\sigma : \sigma \Vdash \psi\}$. By XII.1.6, $S \in \Sigma^0_n$. Let $\sigma \subseteq A$ be such that
$$\sigma \in S \text{ or } (\forall \tau \supseteq \sigma)(\tau \notin S).$$
If $\sigma \in S$, then $A \Vdash \psi$. If $(\forall \tau \supseteq \sigma)(\tau \notin S)$, then $A \Vdash \neg\psi$. So A is n-generic. $\Box$

The next exercise shows that, for $n = 1$, Definitions XII.1.8 and XI.2.1 are consistent.

Exercise XII.1.14 *A is 1-generic if and only if, for every e and x, $\{e\}^A(x)$ is either defined or strongly undefined.* (Hint: for any Σ^0_1 set S of strings, there exist e and x such that S and $\{\sigma : \{e\}^\sigma(x)\downarrow\}$ define the same basic open set in $\mathcal{P}(\omega)$.)

A related characterization of ω-genericity is given by the next result.

Definition XII.1.15 *S is a* **dense set of strings** *if every string has an extension belonging to it.*

Proposition XII.1.16 *A is ω-generic if and only if $\{\sigma : \sigma \subseteq A\}$ meets every dense arithmetical set of strings.*

Proof. Let A be ω-generic and let S be a dense arithmetical set of strings. By density it cannot happen that, for some σ, every extension of σ is not in S. Then, by XII.1.13.2, there is $\sigma \subseteq A$ such that $\sigma \in S$.

Now let A be such that $\{\sigma : \sigma \subseteq A\}$ meets every dense arithmetical set of strings. Given an arithmetical sentence φ, the set $S = \{\sigma : \sigma \Vdash \varphi$ or $\sigma \Vdash \neg\varphi\}$ is dense (by quasi-completeness) and arithmetical (by definability of forcing), and thus there is $\sigma \subseteq A$ such that $\sigma \Vdash \varphi$ or $\sigma \Vdash \neg\varphi$. Then A is ω-generic. $\square$

Exercises XII.1.17 Weak $n+1$-genericity. (Kurtz [1983]) A is **weakly $n+1$-generic** if $\{\sigma : \sigma \subseteq A\}$ meets every dense Σ^0_{n+1} set of strings.

a) *A is weakly $n+1$-generic if and only if $\{\sigma : \sigma \subseteq A\}$ meets every dense Π^0_n set of strings.* (Hint: if $\sigma \in S \Leftrightarrow \exists x R(x,\sigma)$ with $R \in \Pi^0_n$, consider the set

$$\tau \in T \Leftrightarrow (\exists \sigma \subseteq \tau)(\exists x < |\tau|)R(x,\sigma).)$$

b) *A weakly $n+1$-generic set is neither Σ^0_{n+1} nor Π^0_{n+1}.*

c) *A degree contains a weakly 1-generic set if and only if it is hyperimmune.* (Hint: if A is weakly 1-generic and $A = \{a_0 < a_1 < \cdots\}$, we show that $f(n) = a_n$ is not majorized by any recursive function g, so that A is hyperimmune. Consider the set of strings σ such that, for the finite set $\{b_0 < b_1 < \cdots\}$ coded by σ, $g(n) < b_n$ for some n. This is a Σ^0_1 set of strings, hence A meets it. Thus $g(n) < a_n$ for some n. Conversely, let B be hyperimmune and let $g \leq_T B$ not be majorized by any recursive function. We want to find a weakly 1-generic set A such that $A \equiv_T B$. Let $\{S_e\}_{e\in\omega}$ be a uniform enumeration of the Σ^0_1 sets of strings. Build A by finite initial segments σ_s such that $|\sigma_s| \leq s$, recursively in B, so that $A \leq_T B$. And code B into A, so that $B \leq_T A$. At stage $s+1$ do nothing unless there is $e \leq s$ such that the requirement 'S_e meets A if dense' is still unsatisfied, but there is σ such that $\sigma \in S_{e,g(s)}$ and $\sigma \supseteq \sigma_s * 0 * 1^{e+1} * 0$ and $|\sigma| \leq s$. Then take the least such e, the smallest relative σ, and let $\sigma_{s+1} = \sigma * \langle B(n)\rangle$, for the least n not yet coded into A. $A \leq_T B$ by construction. A is weakly 1-generic since, given e, if S_e is dense then

$$h(s) = \mu t.(\forall \tau)(|\tau| \leq s \Rightarrow (\exists \sigma \in S_{e,t})(\sigma \supseteq \tau * 0 * 1^{e+1} * 0))$$

is recursive, and does not majorize g. Thus S_e can be met when it gets highest priority. Finally, to obtain $B \leq_T A$, it is enough to recover $|\sigma_{s+1}|$ recursively in A, for the stage $s+1$ in which something is coded into A. Having $|\sigma_{s+1}|$, get A up to it. By looking at its extension $0 * 1^{e+1} * 0$ on A, we obtain the index of the next requirement which is satisfied, i.e. the string σ used in the construction. Then $|\sigma|+1$ is the length of the next string added.)

d) *$n+1$-generic $\Rightarrow$ weakly $n+1$-generic $\Rightarrow$ n-generic, but no implication is reversible.* (Hint: for the second implication, given $S \in \Sigma^0_n$, consider

$$\sigma \in S^* \Leftrightarrow \sigma \in S \vee (\forall \tau \supseteq \sigma)(\tau \notin S).$$

S^* is Σ^0_{n+1} and dense. Part c) gives a counterexample to the first implication, since every nonzero degree below $\mathbf{0}'$ is hyperimmune (V.5.3.d), and there are minimal degrees below $\mathbf{0}'$ (XI.4.5), while no 1-generic degree is minimal by XI.2.3. Part b) gives a counterexample to the second implication, by XII.1.11.2.)

See Demuth [1987], Demuth and Kučera [1987] for more on weak 1-genericity.

Notice that, by part a), a set A is weakly 1-generic if and only if $\{\sigma : \sigma \subseteq A\}$ meets every dense recursive set of strings. Actually, by the Normal Form Theorem, nothing less than weak 1-genericity is obtained by meeting, for example, only dense primitive recursive sets of strings. Thus, to get a notion of genericity compatible with recursive enumerability one has to change the approach more radically. Various notions of **r.e. genericity** have been proposed, see Ingrassia [1981], Maass [1982], Jockusch [1985].

The next result will be useful to connect forcing and degree-theory.

Proposition XII.1.18 (Jockusch [1981]) *A is ω-generic if and only if it belongs to every arithmetical comeager set.*

Proof. Let A be ω-generic, $\mathcal{A}$ be a comeager arithmetical set and let φ be an arithmetical sentence such that

$$X \models \varphi \;\Leftrightarrow\; X \in \mathcal{A}.$$

By genericity, $A \Vdash \varphi$ or $A \Vdash \neg\varphi$. In the first case, $A \models \varphi$ and $A \in \mathcal{A}$. We show that the second case cannot happen. If $A \Vdash \neg\varphi$, there is $\sigma \subseteq A$ such that $\sigma \Vdash \neg\varphi$. Since the set of ω-generic sets is comeager by XII.1.11, its intersection with $\mathcal{A}$ is comeager, and hence dense (V.3.9.a). Thus there is an ω- generic set $B \supseteq \sigma$ in $\mathcal{A}$. But B forces $\neg\varphi$ because so does σ, and hence $B \models \neg\varphi$, i.e. $B \notin \mathcal{A}$, which is a contradiction.

Conversely, if A is in every arithmetical comeager set, in particular A is in every arithmetical open dense set, hence in every

$$\{X : X \Vdash \varphi \text{ or } X \Vdash \neg\varphi\}$$

for any arithmetical sentence φ. Hence A is ω-generic. $\Box$

This last characterization of genericity accounts for the name 'generic': since a generic set is in all relevant comeager sets, it avoids all the properties which are atypical, in the sense that they are possessed only by a small number of sets (a meager one), and it has instead all properties which are typical, in the sense that they are possessed by a large number of sets (a comeager one).

An alternative approach to forcing $\star$

Since the notion of genericity can be defined independently of forcing, one might think of using the former as a starting point for an alternative approach to the latter. The idea is to *define*

$$\sigma \Vdash^w \varphi \Leftrightarrow \text{ for every generic set } A \text{ such that } A \supseteq \sigma,\ A \models \varphi.$$

The exercises show that this provides an equivalent formulation of weak forcing.

Exercises XII.1.19 a) *For any string σ and any Σ^0_n or Π^0_n sentence φ, $\sigma \Vdash^w \varphi$ if and only if, for every n-generic set $A \supseteq \sigma$, $A \models \varphi$.* (Hint: if $\sigma \Vdash^w \varphi$ and $A \supseteq \sigma$ is n-generic, $A \Vdash \neg\neg\varphi$ and thus $A \models \varphi$. If $(\sigma \Vdash^w \varphi)$ does not hold, then $\tau \Vdash \neg\varphi$ for some $\tau \supseteq \sigma$. By XII.1.11, there is $A \supseteq \tau$ which is n-generic, so $A \supseteq \sigma$ and $A \models \neg\varphi$.)

b) *For any string σ and any arithmetical sentence φ, $\sigma \Vdash^w \varphi$ if and only if, for every ω-generic set $A \supseteq \sigma$, $A \models \varphi$.*

c) *If A is not ω-generic then there is an arithmetical sentence φ such that $A \Vdash \varphi$ and $A \models \neg\varphi$.* (Hint: choose ψ such that A does not force ψ or $\neg\psi$, and let φ be $\neg\neg(\exists\sigma \subseteq X)(\sigma \Vdash \psi \vee \neg\psi)$. Since $(\exists\sigma \subseteq X)(\sigma \Vdash \psi \vee \neg\psi)$ is true of every ω-generic set, by part b) every string weakly forces it.)

Having the notions of genericity and weak forcing available, one can then prove the Generic Existence Theorem by meeting all dense sets of strings considered in the definition of genericity (which is possible, because there are only countably many of them). One could then turn to a proof that forcing equals truth for generic sets (XII.1.10). Note that consideration of weak forcing is not a theoretical obstacle, since the truth of φ is obviously equivalent to the truth of $\neg\neg\varphi$. But it is a practical one, since there is no connection between its definition and the definition of truth. One is thus led to reintroduce forcing to mimic truth, and *prove* that:

- a generic set forces any given sentence or its negation
- $\sigma \Vdash^w \varphi$ if and only if $\sigma \Vdash \neg\neg\varphi$.

This approach, which is equivalent to the one we followed, is due to Shoenfield [1971] and is frequently adopted in Set Theory, see e.g. Kunen [1980]. The disadvantage of weak forcing for our purposes is that its definition uses a higher level of quantifier complexity.

History of the notion of forcing $\star$

The first appearance of the notion of **genericity** is in Spector [1956] and Friedberg [1957b] (see V.2.21, V.2.24 and V.2.26), in order to control the jump of a set A. Here, the problem is that finite information on A is sufficient to determine whether a computation $\{e\}^A(x)$ converges but not, in general, whether it diverges. One thus builds A such that

$$\{e\}^A(x)\uparrow \Leftrightarrow (\exists \sigma \subseteq A)(\forall \tau \supseteq \sigma)(\{e\}^\tau(x)\uparrow).$$

In our terminology, one builds a 1-generic set A. Since the jump of A is a complete $\Sigma_1^{0,A}$ set, finite information on A actually controls the truth of all one-quantifier sentences in A.

The first appearance of the interpretation of logical connectives and quantifiers given in the definition of the **forcing relation** is in Kreisel [1958]. The forcing rules are *proved* for incompletely defined objects in intuitionism (so-called absolutely free choice sequences), using only two of their relevant properties: assertions about incompletely defined objects must use only continuous information on them (here continuous means compact and monotone, as in Section II.3), and numerical functionals of incompletely defined object variables must be uniformly continuous.

This connection with intuitionism is quite revealing. First, it explains the reason why the forcing relation does not respect logical equivalence (φ and $\neg\neg\varphi$ are not intuitionistically equivalent). Second, it permits us to see weak forcing as a counterpart of the double-negation interpretation of intuitionism in classical logic (see Gödel [1933a]). Finally, it suggests uses of forcing for the interpretation of intuitionism (Kripke [1965]).

The forcing relation and the notion of a generic set were rediscovered by Cohen [1963] in the more complicated set-theoretical framework, and are only two of the many ingredients of what is today called the **method of forcing** in Set Theory. Its first use by Cohen was to prove the independence of the Axiom of Choice and of the Continuum Hypothesis from *ZFC* (see Volume III). The early history of the method is in Moore [1988].

Product forcing $\star$

We now introduce two strengthenings of the notion of genericity. The first is simply a relativization to a given set B, in which we extend the language $\mathcal{L}$ considered in XII.1.1 to a new language $\mathcal{L}_B$, obtained by adding to $\mathcal{L}$ a new set constant that has to be interpreted as B (and which we still denote by B). We thus obtain the notion of A being **ω-generic over B**. All the results proved so far hold relativized to B. In particular, XII.1.16 is extended as follows.

Proposition XII.1.20 *A is ω-generic over B if and only if $\{\sigma : \sigma \subseteq A\}$ meets every dense set of strings arithmetical in B.*

A second generalization consists of considering a language $\mathcal{L}'$ similar to the language $\mathcal{L}$ of XII.1.1, where we have two set constants X_1 and X_2 instead of just one X. This leads to the notion of a **generic pair** (A, B), where A and B are the interpretations of X_1 and X_2, respectively. This is usually referred to as the notion of **product forcing**, and once again all the results proved so far hold for it. In particular, XII.1.16 is extended as follows.

Definition XII.1.21 *S is a* **dense set of pairs of strings** *if every pair of strings has a pointwise extension belonging to it, i.e. if for every (τ_1, τ_2) there is $(\sigma_1, \sigma_2) \in S$ such that $\tau_i \subseteq \sigma_i$ $(i = 1, 2)$.*

Proposition XII.1.22 *(A, B) is ω-generic if and only if*

$$\{(\sigma_1, \sigma_2) : \sigma_1 \subseteq A \ \wedge \ \sigma_2 \subseteq B\}$$

meets every dense arithmetical set of pairs of strings.

The next result connects the two extensions of the notion of forcing introduced above.

Proposition XII.1.23 *For any A and B, the following are equivalent:*

1. *(A, B) is ω-generic*
2. *A is ω-generic and B is ω-generic over A*
3. *B is ω-generic and A is ω-generic over B*
4. *A is ω-generic over B and B is ω-generic over A*
5. *$A \oplus B$ is ω-generic.*

Proof. It is enough to prove the equivalence of parts 1 and 2, part 3 being symmetric to part 2, part 4 being equivalent to the conjunction of parts 2 and 3, and part 5 being obviously equivalent to part 1.

Let (A, B) be ω-generic. To prove that A is ω-generic, we show that any dense arithmetical set of strings S is met by $\{\sigma_1 : \sigma_1 \subseteq A\}$. Consider

$$S^* = \{(\sigma_1, \sigma_2) : \sigma_1 \in S\}.$$

S^* is a dense arithmetical set of pairs of strings, and thus for some $\sigma_1 \subseteq A$ and $\sigma_2 \subseteq B$ we have $(\sigma_1, \sigma_2) \in S^*$ and hence $\sigma_1 \in S$.

To prove that B is ω-generic over A, we show that any dense set of strings S, arithmetical in A, is met by $\{\sigma_2 : \sigma_2 \subseteq B\}$. Since S is arithmetical in A, there is a formula '$\sigma_2 \in S$' in the language $\mathcal{L}_A$ which expresses the fact that $\sigma_2 \in S$, i.e. such that

$$\sigma_2 \in S \ \Leftrightarrow \ A \models \text{`}\sigma_2 \in S\text{'}.$$

As a first approach, consider

$$S^* = \{(\sigma_1, \sigma_2) : \sigma_1 \Vdash `\sigma_2 \in S\text{'}\}.$$

S^* is arithmetical by XII.1.6. If we could prove that S^* is dense, we would obtain by ω-genericity of (A, B) two strings $\sigma_1 \subseteq A$ and $\sigma_2 \subseteq B$ such that $\sigma_1 \Vdash$ '$\sigma_2 \in S$'. Since A is ω-generic by the first part of the proof, we would then obtain $A \models$ '$\sigma_2 \in S$' and thus $\sigma_2 \in S$, as required.

The problem is to show that S^* is dense. Given (τ_1, τ_2), by density of S there is $\sigma_2 \supseteq \tau_2$ such that $\sigma_2 \in S$, so there is $\sigma_1 \subseteq A$ such that $\sigma_1 \Vdash$ '$\sigma_2 \in S$'. But there is no reason to suppose that $\sigma_1 \supseteq \tau_1$. We thus modify our approach a little.

Since S is arithmetical in A, there is a formula 'S is dense' in the language $\mathcal{L}_A$, e.g. $(\forall\sigma)(\exists\tau \supseteq \sigma)$('$\tau \in S$'), which expresses the fact that S is dense, i.e. such that

$$S \text{ is dense} \Leftrightarrow A \models `S \text{ is dense'}$$

Since S is dense, $A \models$ 'S is dense', and since A is ω-generic, there is $\sigma \subseteq A$ such that $\sigma \Vdash$ 'S is dense'. Now consider

$$S^* = \{(\sigma_1, \sigma_2) : \sigma_1 \supseteq \sigma \wedge \sigma_1 \Vdash `\sigma_2 \in S\text{'}\}.$$

S^* is arithmetical and dense above $(\sigma, \emptyset)$. Indeed, given (τ_1, τ_2) such that $\tau_1 \supseteq \sigma$, let $C \supseteq \tau_1$ be ω-generic. Then $C \supseteq \sigma$ and $C \models$ 'S is dense', so there is $\sigma_2 \supseteq \tau_2$ such that $C \models$ '$\sigma_2 \in S$'. By 'Forcing = Truth', there is $\sigma_1 \subseteq C$ such that $\sigma_1 \Vdash$ '$\sigma_2 \in S$', and we may suppose $\sigma_1 \supseteq \tau_1$ because $\tau_1 \supseteq C$ (i.e. σ_1 and τ_1 are compatible).

Since S^* is arithmetical and dense above $(\sigma, \emptyset)$, and $A \supseteq \sigma$, there exist $\sigma_1 \subseteq A$ and $\sigma_2 \subseteq B$ such that $(\sigma_1, \sigma_2) \in S^*$, in particular $\sigma_1 \Vdash$ '$\sigma_2 \in S$', and so $\sigma_2 \in S$. Thus S is met, and B is ω-generic over A.

Finally, let A be ω-generic, and B be ω-generic over A. Take a dense arithmetical set S of pairs of strings. Consider

$$S^* = \{\sigma_2 : (\exists\sigma_1 \subseteq A)((\sigma_1, \sigma_2) \in S)\}.$$

S^* is arithmetical in A. To prove it is dense, fix τ_2. Then

$$S^{**} = \{\sigma_1 : (\exists\sigma_2 \supseteq \tau_2)((\sigma_1, \sigma_2) \in S)\}$$

is dense and arithmetical. By ω-genericity of A, there is a string $\sigma_1 \subseteq A$ in S^{**}, and hence S^* is dense. By ω-genericity of B over A, there is a string $\sigma_2 \subseteq B$ in S^*. Thus S is met and (A, B) is ω-generic. $\Box$

Local forcing on trees

In applications, one does not always want to force *all* sentences of a given complexity, but only a certain subset of them. The definition of finite forcing as given in XII.1.2 is not very flexible in this respect, and one needs a notion of relative genericity, with respect only to a given subset of arithmetical sentences. This can easily be achieved by relativizing the notion of forcing itself.

The idea is very simple and it amounts to thinking of a tree T as if it were the identity tree (i.e. the set of all finite strings), and redefining finite forcing on it.

Definition XII.1.24 (Sacks [1971]) *Let $\mathcal{L}$ be the language considered in XII.1.1. If T is a tree, φ is an arithmetical sentence and $\sigma \in T$, then $\boldsymbol{\sigma \Vdash^T \varphi}$ ($\boldsymbol{\sigma}$ **forces** $\boldsymbol{\varphi}$ **on** $\boldsymbol{T}$) is defined by induction on φ as follows:*

$$\begin{array}{lcl}
\sigma \Vdash^T \varphi_R(\overline{n}_1, \dots, \overline{n}_m) & \Leftrightarrow & R(n_1, \dots, n_m) \\
\sigma \Vdash^T \overline{n} \in X & \Leftrightarrow & \sigma(n) = 1 \\
\sigma \Vdash^T \neg\varphi & \Leftrightarrow & (\forall \tau \supseteq \sigma)[\tau \in T \Rightarrow \textit{not } (\tau \Vdash^T \varphi)] \\
\sigma \Vdash^T \varphi_0 \vee \varphi_1 & \Leftrightarrow & \sigma \Vdash^T \varphi_0 \textit{ or } \sigma \Vdash^T \varphi_1 \\
\sigma \Vdash^T (\exists x)\varphi(x) & \Leftrightarrow & \textit{for some } n, \sigma \Vdash^T \varphi(\overline{n}).
\end{array}$$

Note that $\sigma \Vdash^T \varphi$ is *not* equivalent to $\sigma \in T \;\wedge\; \sigma \Vdash \varphi$, because of the definition of $\sigma \Vdash^T \neg\varphi$.

Definition XII.1.25 *Given A on T, then:*

- $\boldsymbol{A \Vdash^T \varphi}$ *if, for some $\sigma \subseteq A$, $\sigma \Vdash^T \varphi$.*
- *A is* $\boldsymbol{n}$**-generic on** $\boldsymbol{T}$ *if, for every sentence $\varphi \in \Sigma^0_n$, $A \Vdash^T \varphi$ or $A \Vdash^T \neg\varphi$.*
- *A is* $\boldsymbol{\omega}$**-generic on** $\boldsymbol{T}$ *if, for every arithmetical sentence φ, $A \Vdash^T \varphi$ or $A \Vdash^T \neg\varphi$.*

The results on usual forcing may be easily localized to a tree T, either by the same proofs or by an appeal to the homeomorphism between T and the full binary tree. The only one worth mentioning is the following.

Proposition XII.1.26 Definability of local forcing (Sacks [1971]) *If T is arithmetical, then:*

1. *$\{(\sigma, \varphi) : \varphi \in \Sigma^0_n \;\wedge\; \sigma \Vdash^T \varphi\}$ is arithmetical*
2. *$\{(\sigma, \varphi) : \varphi$ is arithmetical $\;\wedge\; \sigma \Vdash^T \varphi\}$ is recursive in $\emptyset^{(\omega)}$.*

Proof. By relativization of XII.1.6, the set in part 1 is recursive in $T^{(n)}$ and the set in part 2 is recursive in $T^{(\omega)}$. If T is arithmetical, then $T^{(n)}$ is arithmetical and $T^{(\omega)} \equiv_T \emptyset^{(\omega)}$. $\Box$

Note that forcing of $n+1$-quantifier sentences can now be obtained by locally forcing 1-quantifier sentences on a tree of n-generic sets, and thus n-genericity is really an iteration of 1-genericity, of course applied at each level to a tree of increasing complexity. The advantage of local forcing is, however, that it can be applied to a restricted class of sentences.

Exercises XII.1.27 Sacks forcing in arithmetic. (Sacks [1971]) In the following, T and Q are arithmetical trees and φ is an arithmetical formula. Define a forcing relation with arithmetical trees as conditions, as follows:

$$\begin{array}{lcl} T \Vdash \varphi & \Leftrightarrow & (\forall A \in T)(A \models \varphi) \\ T \Vdash^w \varphi & \Leftrightarrow & (\forall Q \subseteq T)(\exists R \subseteq Q)(R \Vdash \varphi) \\ A \Vdash \varphi & \Leftrightarrow & \text{for some } T,\ A \in T \wedge T \Vdash \varphi. \end{array}$$

A is **Sacks ω-generic** if, for every arithmetical sentence φ, $A \Vdash \varphi$ or $A \Vdash \neg\varphi$.

a) *Monotonicity: if $T \Vdash \varphi$ and $Q \subseteq T$, then $Q \Vdash \varphi$.*

b) *Consistency: $T \Vdash \varphi$ and $T \Vdash \neg\varphi$ cannot both hold.*

c) *Quasi-completeness:* $(\forall T)(T \Vdash \neg\varphi \vee (\exists Q \subseteq T)(Q \Vdash \varphi))$. (Hint: given T and $\varphi \in \Sigma^0_n$, find σ such that $\sigma \Vdash^T \varphi$ or $\sigma \Vdash^T \neg\varphi$, and let Q be a tree of n-generic sets on T extending σ.)

d) *If A is Sacks ω-generic, then $A \models \varphi$ if and only if $A \Vdash \varphi$.*

e) *Every tree T contains a subtree of Sacks ω-generic sets.* (Hint: use quasi-completeness.)

f) *There is a Sacks ω-generic set recursive in $\emptyset^{(\omega)}$.*

g) *Sacks ω-generic sets are not arithmetical.* (Hint: they are not implicitly definable in arithmetic.)

h) *A Sacks ω-generic set can contain an infinite arithmetical subset.* (Hint: use part e) for an appropriate T.)

i) *Fusion Lemma: if $\{\varphi_n\}_{n\in\omega}$ is an arithmetical set of arithmetical sentences of bounded complexity, and $T \Vdash^w \varphi_n$ for all n, then there is a subtree Q of T such that $Q \Vdash \varphi_n$, for all n.* (Hint: let m be such that $\varphi_n \in \Sigma^0_m$ for all n. Let Q be a tree of m-generic sets on T. Suppose Q does not force φ_n; then $A \not\models \varphi_n$ for some A on Q, so $A \models \neg\varphi_n$ and $A \Vdash^T \neg\varphi_n$. Let $\sigma \in T$ be such that $\sigma \Vdash^T \neg\varphi_n$ and let T be a tree of m-generic sets on T above σ. Then $T \models \neg\varphi_n$, contradicting $T \Vdash^w \varphi_n$ and consistency.)

l) *The Fusion Lemma implies quasi-completeness.* (Hint: by induction on the complexity of φ.)

XII.2 Applications of Forcing

The applications of the method of forcing are extremely varied and in this section we only touch on the most immediate ones. First we deal with reducibilities, starting with Turing degrees and then considering nonstandard arithmetical reducibilities. Next we turn to questions of arithmetical definability, both explicit and implicit. Finally, we introduce ω-hops, which are generalizations of the ω-jump operator analogous to the generalizations of the jump operator given by single hops (Section XI.1), and with similarly rich applications (in particular, to the definability of the Turing degrees of arithmetical sets, in XII.3.14).

Turing degrees

ω-generic sets have various interesting degree-theoretical properties. Basically, by XII.1.18, all those that can be proved to be nonempty by a finite extension argument. The next two results consider properties related to Turing reducibility and the jump operator, respectively.

Proposition XII.2.1 (Jockusch [1981]) *If A is ω-generic and $A \in \boldsymbol{a}$, then $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ has the following properties:*

1. *it embeds every countable partial ordering*
2. *it is not a lattice*
3. *it does not have minimal elements*
4. *it is not dense.*

Proof. Parts 1, 2 and 3 follow from the fact that all the relevant comeager sets built in Chapter V (see V.2.9, V.4.9.b, V.3.17) are arithmetical, and by applying XII.1.18 to them.

Part 4 follows from the fact (X.2.9) that the set

$$\{\boldsymbol{a} : (\exists \boldsymbol{b})(\boldsymbol{b} < \boldsymbol{a} \ \wedge\ \boldsymbol{a} \text{ is r.e. in } \boldsymbol{b})\}$$

is comeager and arithmetical. Thus, if $\boldsymbol{a}$ is ω-generic, $\boldsymbol{a}$ is r.e. in a smaller degree $\boldsymbol{b}$. By relativization of XI.4.9, $\boldsymbol{b}$ has a minimal cover below $\boldsymbol{a}$. □

Exercises XII.2.2 a) *If $\boldsymbol{a}$ and $\boldsymbol{b}$ are ω-generic, then the structures $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{a})$ and $\boldsymbol{\mathcal{D}}(\leq \boldsymbol{b})$ are elementarily equivalent.* (Jockusch [1981]) (Hint: for any sentence φ in the language of partial orderings, the property 'the principal ideal defined by the degree of A satisfies φ' is arithmetical and finitely invariant. Hence it is satisfied by all or no ω-generic sets.) Shore [1982b] has shown that *the two structures are not always isomorphic*. Compare this with XI.2.6.a.

b) *If $\boldsymbol{a}$ is ω-generic, then there is a nonzero degree $\boldsymbol{b} < \boldsymbol{a}$ which is not ω-generic.* (Jockusch [1981]) (Hint: $\boldsymbol{a}$ is ω-generic, hence 1-generic and $\mathbf{GL}_1$. There is $\boldsymbol{b} < \boldsymbol{a}$ such that $\boldsymbol{a}$ is r.e. in $\boldsymbol{b}$ and $\boldsymbol{a} \not\leq \boldsymbol{b} \cup \mathbf{0}'$, by relativization to $\mathcal{K}$ of the fact that the set of sets which are r.e. in sets of strictly smaller degree is comeager and arithmetical. Since $\boldsymbol{a} \leq \boldsymbol{b}'$, it cannot be $\boldsymbol{b}' \leq \boldsymbol{b} \cup \mathbf{0}'$, so $\boldsymbol{b}$ is not even 1-generic.) Compare this with XI.2.6.b.

c) *There is a meager set of nonzero degrees whose upward closure is not meager, and is actually comeager*. (Hint: the class of degrees which are not ω-generic is such, by part b.) Compare this with V.3.16.

For more on the structure of degrees of ω-generic sets, see Kumabe [1993], [1993a], [1996].

We turn now to applications of genericity in the study of iterations of the jump operator.

Proposition XII.2.3 Jump properties of generic sets.

1. *If A is n-generic, then $A^{(n)} \equiv_T A \oplus \emptyset^{(n)}$*
2. *If A is ω-generic, then $A^{(\omega)} \equiv_T A \oplus \emptyset^{(\omega)}$.*

Proof. Since $A \oplus \emptyset^{(n)} \leq_T A^{(n)}$ always holds, we prove the reverse inequality for A ω-generic. Let $\varphi(x)$ be '$x \in X^{(n)}$'. Then $\varphi \in \Sigma^0_n$, and by 'forcing equals truth for generic sets' (XII.1.10)

$$x \in A^{(n)} \;\Leftrightarrow\; A \models \varphi(\overline{x}) \;\Leftrightarrow\; (\exists \sigma \subseteq A)(\sigma \Vdash \varphi(\overline{x})).$$

By definability of forcing, $\sigma \Vdash \varphi(\overline{x})$ is Σ^0_n, and thus the right-hand side is r.e. in $A \oplus \emptyset^{(n)}$. Similarly,

$$x \notin A^{(n)} \;\Leftrightarrow\; A \models \neg\varphi(\overline{x}) \;\Leftrightarrow\; (\exists \sigma \subseteq A)(\sigma \Vdash \neg\varphi(\overline{x})).$$

By definability of forcing, $\sigma \Vdash \neg\varphi(\overline{x})$ is Π^0_n, and the right-hand side is again r.e. in $A \oplus \emptyset^{(n)}$. By Post's Theorem II.1.19 relativized, $A^{(n)} \leq_T A \oplus \emptyset^{(n)}$.

Part 2 follows from the uniformity of the above proof. $\Box$

We can now use ω-genericity in the same way as 1-genericity was used to prove results about the jump, to obtain similar results for the ω-jump.

Proposition XII.2.4 $\boldsymbol{\omega}$-Jump Inversion Theorem (MacIntyre [1977]) *The range of the ω-jump operator is the cone $\boldsymbol{\mathcal{D}}(\geq \mathbf{0}^{(\boldsymbol{\omega})})$.*

Proof. Similar to V.2.24, by building an ω-generic set coding a given set C such that $\emptyset^{(\omega)} \leq_T C$. Concisely, build a tree T recursive in $\emptyset^{(\omega)}$, all of whose branches are ω-generic. If $\emptyset^{(\omega)} \leq_T C$, follow the branch $A = \bigcup_{\sigma \subseteq C} T(\sigma)$ indicated by C. Then $A \oplus \emptyset^{(\omega)} \equiv_T C$ by construction and $A^{(\omega)} \equiv_T A \oplus \emptyset^{(\omega)}$ by ω-genericity. $\square$

Because of the use of ω-generic sets in the proof, we have actually proved that *every degree $\boldsymbol{c} \geq \mathbf{0}^{(\boldsymbol{\omega})}$ is the ω-jump of a degree realizing the least possible ω-jump.*

Exercises XII.2.5 a) *The range of the n-jump operator is the cone $\boldsymbol{\mathcal{D}}(\geq \mathbf{0}^{(\boldsymbol{n})})$.* (Selman [1972]) (Hint: use n-generic sets. Or relativize the result for the jump operator.)

b) *The ω-jump and the n-jump operators are never one-one on their ranges.* (Hint: see V.2.22 or V.2.26.)

Arithmetical reducibilities $\star$

The relations $\leq_{\Delta_n^0}$ $(n > 1)$ are not transitive (IV.1.20), hence they do not induce notions of degree. We may nevertheless lift results about Turing degrees to them, by using forcing as the main new tool in proofs. We just give some examples, since we prefer to devote our attention to standard reducibilities (see the next chapters). We prove analogues of V.2.2 and X.1.7.

Proposition XII.2.6 (Selman [1972]) *For each $n \geq 1$, there are Δ_{n+1}^0 sets A and B which are Δ_n^0-incomparable, i.e. neither is Δ_n^0 in the other.*

Proof. It is easier to consider only Σ_n^0 reductions (and this will actually produce a stronger result, namely Σ_n^0-incomparability). We thus build A and B such that, for each $\varphi \in \Sigma_n^0$,

- A is not Σ_n^0-reducible to B via φ, i.e.

$$(\exists x)[x \notin A \Leftrightarrow B \models \varphi(\overline{x})]$$

- B is not Σ_n^0-reducible to A via φ, i.e.

$$(\exists x)[x \notin B \Leftrightarrow A \models \varphi(\overline{x})].$$

For example, suppose we want to satisfy the first condition. We make B n-generic, so that

$$B \models \varphi(\overline{x}) \Leftrightarrow (\exists \sigma \subseteq B)(\sigma \Vdash \varphi(\overline{x})).$$

It is thus enough to obtain x such that

$$x \notin A \Leftrightarrow (\exists \sigma \subseteq B)(\sigma \Vdash \varphi(\overline{x})),$$

and this we can easily do by the finite extension method. Similarly, to satisfy the second condition, we make A n-generic.

Consider recursive enumerations $\{\psi_e\}_{e\in\omega}$ and $\{\varphi_e\}_{e\in\omega}$ of the Σ^0_n sentences and formulas with one free variable, respectively. We let $A = \bigcup_{s\in\omega} \sigma_s$ and $B = \bigcup_{s\in\omega} \tau_s$. We start with $\sigma_0 = \tau_0 = \emptyset$. At stage $s+1$, we have σ_s and τ_s, and proceed as follows:

- $s = 2e$
  Let $\sigma \supseteq \sigma_s$ be an initial segment such that $\sigma \Vdash \psi_e$ or $\sigma \Vdash \neg\psi_e$, which exists by quasi-completeness. Then let x be the smallest element such that $\sigma(x)$ does not converge. See whether there is an initial segment $\tau \supseteq \tau_s$ such that $\tau \Vdash \varphi_e(\overline{x})$.

  If so, take the least such τ and let $\tau_{s+1} = \tau$ and $\sigma_{s+1} = \sigma * 0$. Then $x \notin A$ and $B \Vdash \varphi_e(\overline{x})$.

  If not, let $\tau_{s+1} = \tau_s$ and $\sigma_{s+1} = \sigma * 1$, so that $x \in A$ but B does not force $\varphi_e(\overline{x})$, otherwise $\tau \Vdash \varphi_e(\overline{x})$ for some $\tau \supseteq \tau_s$.

- $s = 2e + 1$
  The construction is the same in this case, with A and B interchanged.

By construction, A and B are n-generic. By construction and 'forcing equals truth for generic sets' (XII.1.10), they are Σ^0_n-incomparable. It remains to check that A and B are Δ^0_{n+1}. But the most difficult questions we ask are whether $(\exists \tau \supseteq \tau_s)(\tau \Vdash \varphi_e(\overline{x}))$ and $(\exists \sigma \supseteq \sigma_s)(\sigma \Vdash \varphi_e(\overline{x}))$, which are Σ^0_n by definability of forcing, and hence the construction is Δ^0_{n+1}. $\square$

A less direct, alternative proof of the previous result can be given as follows. Define n-generic pairs (A, B) in a way analogous to ω-generic pairs. Note that if (A, B) is such a pair, then A and B are mutually Δ^0_n-incomparable. And note that the natural construction of such a pair produces sets A and B recursive in $\emptyset^{(n)}$, and hence Δ^0_{n+1}.

Proposition XII.2.7 (Hinman [1969]) *For each $n \geq 1$, there are Σ^0_n sets A and B which are Δ^0_n-incomparable, i.e. neither is Δ^0_n in the other.*

Proof. For each $n \geq 0$, we want Σ^0_{n+1} sets A and B which are Δ^0_{n+1}-incomparable. We cannot expect to make them $n+1$-generic, otherwise they could not be Σ^0_{n+1}. What we can do is to make them n-generic, and see that this is enough to control all Δ^0_{n+1} reductions (and not only Σ^0_n ones) by finite strings.

Recall that to say that A is $\Delta^{0,B}_{n+1}$ means that the graph of the characteristic function of A is such. It is enough for c_A to be $\Sigma^{0,B}_{n+1}$ since then, being total, it is automatically $\Delta^{0,B}_{n+1}$:

$$c_A(x) = y \;\Leftrightarrow\; (\exists z)(c_A(x) = z \wedge y \neq z).$$

Let $\{\psi_e\}_{e\in\omega}$ and $\{\varphi_e\}_{e\in\omega}$ be recursive enumerations of the Σ^0_n sentences and the Π^0_n formulas with three free variables, respectively. The proof uses forcing and priority.

1. To satisfy
$$c_A(x) \neq y \;\Leftrightarrow\; B \models (\exists z)\varphi_e(\overline{x}, \overline{y}, z)$$
for some x and y (in case this condition has not yet been satisfied and injured afterwards), note that at the end, since B will be n-generic and $\varphi_e \in \Pi^0_n$,
$$\begin{aligned} & B \models (\exists z)\varphi_e(\overline{x}, \overline{y}, z) \\ \Leftrightarrow\quad & (\exists z)(B \models \varphi_e(\overline{x}, \overline{y}, \overline{z})) \\ \Leftrightarrow\quad & (\exists z)(\exists \sigma \subseteq B)(\sigma \Vdash \varphi_e(\overline{x}, \overline{y}, \overline{z})). \end{aligned}$$
Pick a fresh witness x and, at a stage $s+1$ when the requirement is considered, see if for some $\sigma \subseteq B_s$ with length $\leq s$, and some $z \leq s$, $\sigma \Vdash \varphi_e(\overline{x}, \overline{0}, \overline{z})$.

   If this never happens, then $B \models \neg(\exists z)\varphi_e(\overline{x}, \overline{0}, z)$ and x is never put into A (if injuries do not occur), so $c_A(x) = 0$.

   If this happens at stage $s+1$, put x into A (so that $c_A(x) = 1$), and freeze all numbers z such that $\sigma(z) = 0$: if they are never put into B afterwards, $\sigma \subseteq B$ and $B \models (\exists z)\varphi_e(\overline{x}, \overline{0}, z)$.

2. To satisfy
$$A \Vdash \psi_e \text{ or } A \Vdash \neg\psi_e$$
at stage $s+1$ when it receives attention (in case this condition has not yet been satisfied and injured afterwards), search for σ such that:

   - $\sigma \Vdash \psi_e$ or $\sigma \Vdash \neg\psi_e$
   - σ gives value 1 to the elements of A_s on which it is defined
   - σ gives value 0 to the elements we do not want to put in A, either because they have been frozen when a condition with higher priority has been satisfied, or because they are witnesses of conditions of higher priority (as in part 1) and have not yet been put into A.

Then put into A the elements such that $\sigma(z) = 1$, and freeze out of A those for which $\sigma(z) = 0$. Note that we do not need to do a bounded search in this case, since σ always exists (by quasi-completeness).

There is also going to be action symmetric to both parts 1 and 2, with the roles of A and B interchanged. The proof is a standard finite injury argument, with injuries occurring not only for the sake of making A and B Δ^0_{n+1}-incomparable, but also for the sake of making them n-generic. A and B are Σ^0_{n+1} because they are enumerated recursively in $\emptyset^{(n)}$. □

Exercises XII.2.8 *For $n \geq 1$, if $A \notin \Delta^0_n$, then there is B which is Δ^0_n-incomparable with it.* (Hinman [1969]) (Hint: by adding forcing to V.2.13 and using c_A as above.)

Arithmetical definability ⋆

Another application of forcing is to provide (non)definability results.

Proposition XII.2.9 (Addison [1965]) *The class of arithmetical sets is not arithmetical.*

Proof. Suppose otherwise. For some arithmetical formula φ and every X,

$$X \text{ is arithmetical} \Leftrightarrow X \models \varphi.$$

Let n be such that $\varphi \in \Sigma^0_n$. By XII.1.11, there is A n-generic and arithmetical. Thus $A \models \varphi$ and $\sigma \Vdash \varphi$ for some $\sigma \subseteq A$. But then every n-generic set extending σ satisfies φ and is arithmetical, contradicting the fact that there are comeager (and hence uncountably) many such sets, but only countably many arithmetical ones. □

Exercises XII.2.10 *The class of Δ^0_n sets is not Σ^0_{n+1}.* (Hint: otherwise, for some $\varphi \in \Pi^0_n$, $X \in \Delta^0_{n+1} \Leftrightarrow X \models (\exists x)\varphi(x)$. Then one can apply the same reasoning as above.) A better lower bound on the complexity of these classes is obtained in XII.2.12.

Since the class of arithmetical sets is countable, its complement is comeager (V.3.9.c), and it does not have arithmetical members. We now prove that every comeager (actually, nonmeager) *arithmetical* class of sets must have an arithmetical member, thus providing a different proof of XII.2.9. The complementary case of countable arithmetical classes is dealt with in XII.2.20.

Proposition XII.2.11 Basis Theorem for nonmeager arithmetical classes (Hinman [1969])

1. *Every nonmeager Σ^0_{n+3} class has a Δ^0_{n+1} member.*
2. *Every nonmeager arithmetical class has an arithmetical member.*

Proof. Part 2 follows from part 1, but we prove it independently to isolate the main idea of the proof. Let $\mathcal{A}$ be nonmeager and $\varphi \in \Sigma^0_n$ be such that

$$X \in \mathcal{A} \Leftrightarrow X \models \varphi.$$

First we note that $\mathcal{A}$ has an n-generic member. Indeed, the set $\mathcal{G}_n$ of all n-generic sets is comeager by XII.1.11, so $\mathcal{A} \cap \mathcal{G}_n$ is not meager (otherwise $\mathcal{A} \subseteq (\mathcal{A} \cap \mathcal{G}_n) \cup \overline{\mathcal{G}}_n$ would also be meager), and hence is not empty. Let $A \in \mathcal{A} \cap \mathcal{G}_n$. $A \models \varphi$, because $A \in \mathcal{A}$. And $\sigma \Vdash \varphi$ for some $\sigma \subseteq A$, because $A \in \mathcal{G}_n$. Now let B be an arithmetical n-generic set extending σ. Then $B \models \varphi$ and so $B \in \mathcal{A}$.

We now prove part 1. Let $\mathcal{A} \in \Sigma^0_{n+3}$ be nonmeager and $\varphi \in \Pi^0_n$ be such that

$$\begin{aligned} X \in \mathcal{A} \quad &\Leftrightarrow \quad X \models \exists x \forall s \exists z \varphi(x,s,z) \\ &\Leftrightarrow \quad \text{for some } x,\ X \models \forall s \exists z \varphi(\overline{x},s,z). \end{aligned}$$

For each x, consider

$$\mathcal{A}_x = \{X : X \models \forall s \exists z \varphi(\overline{x},s,z)\}.$$

Since $\mathcal{A} = \bigcup_{x\in\omega} \mathcal{A}_x$ and a countable union of meager sets is meager, $\mathcal{A}_x$ must be nonmeager for some x. So $\mathcal{A}_x \cap \mathcal{G}_n$ is also not meager (where $\mathcal{G}_n$ is the set of n-generic sets, as in the previous proof), in particular is not nowhere dense. Thus there is σ_0 such that $\mathcal{A}_x \cap \mathcal{G}_n$ is dense above σ_0. We build $A \in \mathcal{A}_x \cap \mathcal{G}_n$ recursively in $\emptyset^{(n)}$, as follows. Let $\{\varphi_s\}_{s\in\omega}$ be a recursive enumeration of the Σ^0_n sentences. We start with σ_0, which we already have.

- σ_{2s+1} is the least $\sigma \supseteq \sigma_{2s}$ such that $\sigma \Vdash \varphi_s$ or $\sigma \Vdash \neg\varphi_s$, which exists by quasi-completeness.
- σ_{2s+2} is the least $\sigma \supseteq \sigma_{2s+1}$ such that $\sigma \Vdash \varphi(\overline{x},\overline{s},\overline{z})$ for some z. It exists because $\mathcal{A}_x \cap \mathcal{G}_n$ is dense above σ_0, so there is $B \in \mathcal{A}_x \cap \mathcal{G}_n$ such that $B \supseteq \sigma_{2s+1}$. Since $B \in \mathcal{A}_x$, $B \models \forall s \exists z \varphi(\overline{x},s,z)$, and in particular $B \models \varphi(\overline{x},\overline{s},\overline{z})$ for some z. Since $B \in \mathcal{G}_n$, $B \Vdash \varphi(\overline{x},\overline{s},\overline{z})$ for some z.

If $A = \bigcup_{s\in\omega} \sigma_s$ then A is recursive in $\emptyset^{(n)}$ by construction, n-generic by definition of σ_{2s+1}, and a member of $\mathcal{A}_x$ because the definition of σ_{2s+2} ensures

$$\forall s \exists z (A \Vdash \varphi(\overline{x},\overline{s},\overline{z})),$$

so

$$\forall s \exists z (A \models \varphi(\overline{x}, \overline{s}, \overline{z}))$$

and

$$A \models \forall s \exists z \varphi(\overline{x}, s, z).$$

In particular, $A \in \mathcal{A}$. $\Box$

For arithmetical classes of positive measure, the analogue of part 2 holds (Tanaka [1967], Sacks [1969]) but the analogue of part 1 fails (Tanaka [1970]).

Corollary XII.2.12 (Shoenfield [1958a], Hinman [1969]) *The following classes of sets are $\Sigma^0_{n+3} - \Delta^0_{n+3}$:*

1. $\{A : A \in \Sigma^0_{n+1}\}$
2. $\{A : A \in \Pi^0_{n+1}\}$
3. $\{A : A \in \Delta^0_{n+1}\}$.

Proof. The computations that show that the given classes have the required complexity are routine. For example, for $n = 0$ we have

$$\begin{aligned} A \in \Sigma^0_1 \quad &\Leftrightarrow \quad (\exists e)(A = \mathcal{W}_e) \\ &\Leftrightarrow \quad (\exists e)(\forall x)(x \in A \leftrightarrow x \in \mathcal{W}_e). \end{aligned}$$

Since the condition '$x \in \mathcal{W}_e$' is Σ^0_1, the whole expression is Σ^0_3. Similarly for $n > 0$, and this proves the positive part of 1. The positive part of 2 follows by negation, and that of 3 from the previous parts.

Suppose now one of these classes is Π^0_{n+3}. Since it is countable, its complement is a comeager, and hence a nonmeager Σ^0_{n+3} class of sets. By the previous result it contains a Δ^0_{n+1} member, which is a contradiction. $\Box$

It follows that XII.2.11.1 is best possible, because the complement of the class of Δ^0_{n+1} sets is *a Π^0_{n+3} comeager class that does contain any Δ^0_{n+1} set.*

Implicit arithmetical definability $\star$

The next concept is an extension of (explicit) arithmetical definability with a set parameter (XII.1.1).

Definition XII.2.13 *A set A is* **implicitly definable in arithmetic***, or an* **arithmetical set singleton***, if there is an arithmetical formula φ with a set variable such that A is the unique solution of φ:*

$$X = A \Leftrightarrow X \models \varphi.$$

The set A is a **$\mathcal{C}$-singleton**, *also written as* **$\{A\} \in \mathcal{C}$**, *if $\mathcal{C}$ is a class of arithmetical formulas and $\varphi \in \mathcal{C}$.*

Exercises XII.2.14 Function singletons. A function f is a **$\mathcal{C}$-singleton** if it is the unique solution of a formula in $\mathcal{C}$ with a function variable.

a) *If a function is a Π^0_1-singleton, then its graph is a Π^0_2-singleton.* (Hint: given a Π^0_1 formula φ with a function variable whose unique solution is f, consider the Π^0_1 formula obtained from φ by substituting for every atomic statement $f(x) = y$ an atomic statement $\langle x, y\rangle \in X$, and add to it the Π^0_2 formula stating that X is the graph of a function, where the Π^0_2 form comes from the totality requirement.)

b) *For $n > 1$, if a function is a Π^0_n-singleton, then so is its graph.* (Hint: if $n > 1$, then the condition of totality in part a) is Π^0_n.)

c) *For $n \geq 1$, if a set is a Π^0_n-singleton, then so is its characteristic function.* (Hint: given a Π^0_n formula with a set variable whose unique solution is A, consider the Π^0_n formula obtained from φ by substituting for every atomic statement $x \in X$ an atomic statement $f(x) = 1$, and add to it the Π^0_1 clause saying that f is a characteristic function, i.e. that its values are ≤ 1.)

d) *A degree contains a Π^0_1 function singleton if and only if it contains a Π^0_2 set singleton.* (Jockusch and McLaughlin [1969]) (Hint: one direction follows from part a). Conversely, let A be the unique solution of $\forall x \exists y R(x, y, \hat{c}_X(y))$, and consider $f(x) = \hat{c}_A(\mu y.\, R(x, y, \hat{c}_A(y)))$. Then $f \leq_T A$ by definition. $A \leq_T f$ because the length of $f(x)$ goes to infinity, otherwise only a finite segment of A would be used for every x in R, and A could not be the unique solution. Finally, f is a Π^0_1-singleton because it is the unique solution of

$$\forall x[f(x) \text{ is a sequence number of 0's and 1's} \wedge R(x, |f(x)|, f(x))].)$$

e) *For any $n \geq 2$, a degree contains a Π^0_n function singleton if and only if it contains a Π^0_n set singleton.* (Hint: from parts b) and c).)

The next exercises show that Π^0_2-singletons are the first nontrivial examples of arithmetical singletons, and they will turn out to be an extremely rich class.

Exercises XII.2.15 a) *For every n, every Σ^0_{n+1}-singleton is a Π^0_n-singleton.* (Hint: if A is the unique solution of $\exists x \varphi(x)$, choose a such that $A \models \varphi(\overline{a})$.)

b) *For any $n \geq 1$, every Δ^0_n set is a Π^0_n-singleton.* (Hint: if A is explicitly defined by φ, then it is implicitly defined by

$$(\forall x)[x \in X \leftrightarrow \varphi(x)].$$

c) *A set is a Π^0_1-singleton if and only if it is recursive.* (Jockusch and McLaughlin [1969], Odifreddi [1976]) (Hint: a recursive set is a Π^0_1-singleton by part b). Conversely, we can write a Π^0_1 formula φ with set variable X in the form $(\forall x)R[\hat{c}_X(x)]$. Let A be implicitly defined by φ. To decide whether $n \in A$, consider the full binary tree, and at each level x eliminate all branches above any string σ of length x such that $R(\sigma)$ does not hold. Proceed until a string of length greater than n is found such that all remaining branches extend it. Such a string exists because there is a unique set satisfying φ, and it determines the value of A up to its length, in particular for n.)

d) *For any $n \geq 2$, if A is a Π^0_n-singleton and $A \leq_T B \leq_T A'$, then the degree of B contains a Π^0_n-singleton.* (Jockusch and McLaughlin [1969]) (Hint: since $A \leq_T B$, it is enough to implicitly define $A \oplus B$. Since $B \leq_T A'$, by the Limit Lemma there is $\varphi \in \Delta^0_2$ such that

$$x \in B \ \Leftrightarrow\ A \models \varphi(\overline{x}).$$

Then, if $Ev(X)$ and $Od(X)$ are such that $X = Ev(X) \oplus Od(X)$,

$$X = A \oplus B \ \Leftrightarrow\ Ev(X) = A \wedge (\forall x)[x \in Od(X) \leftrightarrow Ev(X) \models \varphi(\overline{x})].$$

and the right-hand side is Π^0_n because A is a Π^0_n-singleton and φ is Δ^0_2.)

e) *For every n, every degree between $\mathbf{0}^{(n)}$ and $\mathbf{0}^{(n+1)}$ contains a Π^0_2-singleton.* (Hint: by induction, from parts c) and d).)

f) *There exist Π^0_2-singletons of minimal degree, and they are all below $\mathbf{0}'$.* (Jockusch and McLaughlin [1969]) (Hint: the existence follows from XI.4.5 and part b). By XII.2.14.d, if A is a Π^0_2-singleton then there is f such that $f \equiv_T A$, and f is a Π^0_1-singleton. As is part c), f is recursive in any function g that dominates it, by considering a tree such that if σ is at level n on it, then its successors at level $n+1$ are $\sigma * \langle x \rangle$, with $x \leq g(n)$. If A has minimal degree then so does f, and by the proof of XI.2.14 there is a function recursive in $\mathbf{0}'$ that dominates it.)

We now classify arithmetical singletons in the Analytical Hierarchy.

Proposition XII.2.16 *Every arithmetical singleton is Δ^1_1.*

Proof. Let φ be an arithmetical formula such that

$$X = A \ \Leftrightarrow\ X \models \varphi.$$

Then

$$\begin{aligned} x \in A \quad &\Leftrightarrow \quad (\forall X)(X \models \varphi \ \rightarrow\ x \in X) \\ &\Leftrightarrow \quad (\exists X)(X \models \varphi \wedge x \ \in X). \end{aligned}$$

By the complexity of truth (IV.1.12.a), the expression '$X \models \varphi$' is arithmetical in X, and thus the two previous expressions for A are Π^1_1 and Σ^1_1. $\square$

We now use the method of forcing to show that the converse of the above proposition is far from being true.

Proposition XII.2.17 (Feferman [1965])

1. *For every $n \geq 1$, there are sets recursive in $\emptyset^{(n)}$ that are not Π^0_n-singletons.*

2. There are sets recursive in $\emptyset^{(\omega)}$ that are not arithmetical singletons.

Proof. Both parts follow from the Generic Existence Theorem XII.1.11, once we prove that *an n-generic set is not a Π^0_n-singleton*, and hence also that an ω-generic set is not an arithmetical singleton.

Suppose A is n-generic, φ is Π^0_n, and

$$X = A \ \Leftrightarrow\ X \models \varphi.$$

Then $A \models \varphi$, and by n-genericity $A \Vdash \varphi$, i.e. $(\exists \sigma \subseteq A)(\sigma \Vdash \varphi)$. Choose $B \supseteq \sigma$ n-generic and different from A, which is possible by XII.1.11.a. Then $B \Vdash \varphi$ because $B \supseteq \sigma$, and hence $B \models \varphi$ by n-genericity. But then $B = A$ because φ implicitly defines A, contradicting the choice of B. $\square$

It follows from XII.2.15.b and XII.2.17.1 that, for $n \geq 1$, every Δ^0_n is a Π^0_n-singleton, but not every Δ^0_{n+1} set is. The best possible result in this direction is that *not every Σ^0_n (or Π^0_n) set is a Π^0_n-singleton* (Jockusch and McLaughlin [1969], Jockusch and Shore [1984]).

Exercises XII.2.18 (Jockusch and Shore [1984], Simpson [1985])

a) *If A is $n+1$-generic and B is a Π^0_n singleton such that $B \leq_T A$, then B is Δ^0_2.* (Hint: if $B \leq_T A$, then $B \simeq \varphi^A_e$ for some e, and

$$x \in B \ \Leftrightarrow\ \varphi^A_e(x) \simeq 1 \ \Leftrightarrow\ (\exists \sigma \subseteq A)(\varphi^\sigma_e(x) \simeq 1).$$

Consider

$$\psi(x) \Leftrightarrow (\exists \sigma \subseteq X)(\varphi^\sigma_e(x) \simeq 1)$$

and, if A is the unique solution of $\varphi \in \Pi^0_n$,

$$\varphi_0 \ \Leftrightarrow\ \{x : X \models \psi(\overline{x})\} \models \varphi.$$

Then φ_0 is Π^0_n in $\{x : X \models \psi(\overline{x})\}$, which is Σ^0_1, both facts by the definability of truth (IV.1.12). Thus φ_0 is Π^0_{n+1}. If A is $n+1$-generic, then $A \Vdash \varphi_0$ because $A \models \varphi_0$, and so $(\exists \sigma \subseteq A)(\sigma \Vdash \varphi_0)$. To compute B, given x search for $\tau \supseteq \sigma$ such that

$$\tau \Vdash \psi(\overline{x}) \ \text{ or } \ \tau \Vdash \neg\psi(\overline{x}).$$

In the first case $x \in B$, in the second $x \in \overline{B}$. Since ψ is Σ^0_1, the two verifications are Σ^0_1 and Π^0_1, respectively, and hence the search procedure is Δ^0_2.)

b) *If A is ω-generic, and B is an arithmetical singleton such that $B \leq_a A$, then B is arithmetical.*

Of course, XII.2.17.2 is interesting only if there are arithmetical singletons that are not arithmetical. The next result provides a natural example.

Proposition XII.2.19 (Hilbert and Bernays [1939], Kuznekov and Trakhtenbrot [1955]) *$\emptyset^{(\omega)}$ is a Π^0_2-singleton.*

Proof. By definition V.1.10, we have $\emptyset^{(\omega)} = \oplus_{n\in\omega}\emptyset^{(n)}$. Then

$$\begin{aligned} X = \emptyset^{(\omega)} \quad &\Leftrightarrow \quad X^{[0]} = \emptyset \,\wedge\, (\forall n)(X^{[n+1]} = (X^{[n]})') \\ &\Leftrightarrow \quad (\forall z)(\langle 0, z\rangle \notin X) \,\wedge \\ &\qquad\quad (\forall n)(\forall z)(\langle n+1, z\rangle \in X \,\leftrightarrow\, \langle n+1, z\rangle \in (X^{[n]})'). \end{aligned}$$

Since $(X^{[n]})'$ is r.e. in $X^{[n]}$ and hence in X, the whole expression turns out to be $\Pi^{0,X}_2$, as claimed. $\square$

In particular, since $\emptyset^{(\omega)}$ is the set coding the truth definition of arithmetic, the previous result is a partial repair of Tarski's Theorem (p. I.166): *arithmetical truth is not explicitly definable in arithmetic, but it is implicitly definable in it*, at the first nontrivial level.

In Volume III we will extend the chain $\{\emptyset^{(\alpha)}\}_{\alpha\leq\omega}$ to all recursive ordinals α, and show on the one hand that every Δ^1_1 set is recursive in $\emptyset^{(\alpha)}$ for some $\alpha < \omega_1^{ck}$, and on the other hand that every such $\emptyset^{(\alpha)}$ is a Π^0_2-singleton. Thus *the Turing degrees of arithmetical (actually, Π^0_2) singletons are cofinal in the Turing degrees of Δ^1_1 sets*. This shows that the set quantifier hidden in the requirement that a set be implicitly definable is not avoidable, and that XII.2.16 is the best general complexity bound that can be computed for arithmetical singletons.

In a different vein, Chapter XIII deals with the a-degrees of arithmetical (actually, Π^0_2) singletons, and shows that they too have a rich structure. The results are stated there for ω-r.e.a. sets, a particular subclass of Π^0_2-singletons (see XIII.2.7).

In XII.2.11 we proved a Basis Theorem for nonmeager arithmetical classes, and we deal here with the complementary case of countable arithmetical classes. Since $\{A\}$ is a countable arithmetical class when A is an arithmetical singleton, the next result is the best one can hope for.

Proposition XII.2.20 Basis Theorem for countable arithmetical classes (Tanaka [1972])

1. *Every nonempty countable Σ^0_{n+1} class contains a Π^0_n-singleton.*
2. *Every nonempty countable arithmetical class contains an arithmetical singleton.*

Proof. Part 2 follows from part 1. By fixing an element, we can consider Π^0_n classes instead of Σ^0_{n+1} classes. We then prove that a nonempty Π^0_n class $\mathcal{A}$ without Π^0_n-singletons is uncountable.

First, note that it is possible to find a recursive relation R such that

$$X \in \mathcal{A} \,\Leftrightarrow\, \exists g\forall x R(\hat{c}_X(x), \hat{g}(x)),$$

by first writing the Π^0_n definition of $\mathcal{A}$ in prenex form, then by changing blocks of the form $\forall x \exists y$ to blocks of the form $\exists g \forall x$, and by contracting quantifiers when needed.

Now consider the set of pairs of strings

$$(\sigma, \tau) \in \mathcal{B} \Leftrightarrow \exists X \exists g[X \supseteq \sigma \wedge g \supseteq \tau \wedge \forall x R(\hat{c}_X(x), \hat{g}(x))].$$

Since $\mathcal{A}$ is not empty, neither is $\mathcal{B}$. Since $\mathcal{A}$ does not contain Π^0_n-singletons, for every $(\sigma, \tau) \in \mathcal{B}$ it is possible to find pairs of strings $(\sigma_i, \tau_i) \in \mathcal{B}$ $(i = 0, 1)$ such that $\sigma_i \supset \sigma$, $\tau_i \supset \tau$ and σ_0 and σ_1 are incompatible. Otherwise, the set

$$X \in \mathcal{C} \Leftrightarrow \exists g[X \supseteq \sigma \wedge g \supseteq \tau \wedge \forall x R(\hat{c}_X(x), \hat{g}(x))]$$

would define a Π^0_n-singleton belonging to $\mathcal{A}$, contrary to the hypothesis.

By the definition of $\mathcal{B}$, an infinite increasing sequence $(\sigma_n, \tau_n)_{n \in \omega}$ of pairs in it defines a set $X = \bigcup_{n \in \omega} \sigma_n$ and a function g such that

$$\forall x R(\hat{c}_X(x), \hat{g}(x)),$$

in particular $X \in \mathcal{A}$. By the property of $\mathcal{B}$ stated above, there is a tree of such X's, and thus $\mathcal{A}$ is uncountable. $\square$

It is not true in general that *every* member of a countable arithmetical class is an arithmetical singleton (Harrington [1976]), although it is always hyperarithmetical (Harrison [1968], see Volume III).

ω-Hops

The jump operator sends any set X into a set X' r.e. in and above X. Being obtained from an r.e. construction (that of $\mathcal{K}$) by relativization, the jump operator is of the form

$$X \longmapsto X \oplus \mathcal{W}^X_e = H_e(X).$$

The operators H_e may thus be thought of as generalizations of the jump operator, and are called **pseudo-jumps** or **hops**. They share many properties with the jump (see Section XI.1).

The double-jump is a double iteration of the jump. Similarly, two hops may be iterated as well:

$$X \longmapsto H_{e_1}(H_{e_0}(X)) = (X \oplus \mathcal{W}^X_{e_0}) \oplus \mathcal{W}^{X \oplus \mathcal{W}^X_{e_0}}_{e_1}.$$

The same can be done for n-jumps.

The ω-jump is a recursive union of finite iterations of the jump. We thus arrive at the notion of ω-hop, which generalizes the ω-jump in the same way as the notion of hop generalizes the jump.

Definition XII.2.21 (Jockusch and Shore [1984]) *Let f be a recursive function. Then*

$$J(X, f) = \oplus_{n\in\omega} H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X))\cdots))$$

is the **$\boldsymbol{\omega}$-hop** *defined by f.*

In the following, mention of f will be dropped if not required

The reason for the name 'ω-hop' is the same as for the name 'hop'. More precisely, for any ω-hop J and any X one has

$$X \leq_T J(X) \leq_T X^{(\omega)},$$

i.e. an ω-hop always goes up in degree, possibly not strictly, and it can be at most an ω-jump. We will see that some of the properties of the ω-jump actually generalize to any ω-hop.

As a first example, the next result generalizes the Jump Inversion Theorem for the ω-jump (XII.2.4), in the same way as XI.1.16 generalized the Jump Inversion Theorem for the jump (V.2.24).

Theorem XII.2.22 $\boldsymbol{\omega}$-Hops Inversion Theorem (Jockusch and Shore [1984]) *For any ω-hop J and any set C such that $\emptyset^{(\omega)} \leq_T C$, there is a set A such that*

$$J(A) \ \equiv_T \ A \oplus_T \emptyset^{(\omega)} \ \equiv_T \ C.$$

Proof. The ideas are the same as in the usual proofs of Inversion Theorems. In V.2.24 and XII.2.4 we used plain 1-genericity and ω-genericity to control jump and ω-jump, respectively. To control hops, in XI.1.16 we used 'partial' genericity, with respect only to the sentences required for the given hop. To control ω-hops, we need to iterate this approach. The natural tool for this is local forcing (p. 751), which was not used in XII.2.4 because we could directly force all sentences with any number of quantifiers.

Let J be defined by f, i.e.

$$J(X) = \oplus_{n\in\omega} H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X))\cdots)).$$

We define a sequence of trees $\{T_n\}_{n\in\omega}$, with T_n recursive in $\emptyset^{(n)}$, and T_{n+1} a subtree of T_n such that all its branches are generic w.r.t. the n-quantifier sentences of the form

$$x \in H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X))\cdots)).$$

We start with

$$T_0 \ = \ \text{identity tree.}$$

Given T_n, we let the first n levels of T_{n+1} be equal to the first n levels of T_n (i.e. $T_{n+1}(\sigma) = T_n(\sigma)$ if $|\sigma| \leq n$), to avoid a collapse of the intersection of all trees. Then we build above each string at level n, a tree of sets generic w.r.t. all sentences of the form '$x \in H_{f(n)}(X)$', by local forcing on T_n. Precisely, given $T_{n+1}(\sigma) = T_n(\tau)$ with $|\sigma| \geq n$ and for some τ, let $T_{n+1}(\sigma * i)$ $(i = 0, 1)$ be the least string $\alpha \supseteq T_n(\tau * i)$ such that

$$\alpha \Vdash^{T_n} (x \in H_{f(n)}(X)) \quad \text{or} \quad \alpha \Vdash^{T_n} \neg(x \in H_{f(n)}(X)),$$

where $x = |\sigma| - n$. In other words, by starting at level n we successively and orderly force the sentences determining the hop $H_{f(n)}$ or their negations. Inductively, every branch of T_{n+1} is thus generic w.r.t. the sentences

$$x \in H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X))\cdots)).$$

Now let $T = \bigcap_{n \in \omega} T_n$. By the first part of the construction of T_{n+1}, T is nonempty and it is still a tree. By definability of local forcing XII.1.26, every T_n is uniformly recursive in $\emptyset^{(n)}$, and thus T is recursive in $\emptyset^{(\omega)}$. Finally, every branch of T is generic w.r.t. all sentences of the form

$$x \in H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X))\cdots)),$$

for every n.

Given C such that $\emptyset^{(\omega)} \leq_T C$, let A be the branch of T determined by C, i.e. $A = \bigcup_{\sigma \subseteq C} T(\sigma)$. We now prove that A has the desired properties.

1. $J(A) \leq_T A \oplus \emptyset^{(\omega)}$
   By definition,

$$\begin{aligned}\langle n, x\rangle \in J(A) \quad &\Leftrightarrow \quad x \in H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(A))\cdots))\\ &\Leftrightarrow \quad A \models x \in H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X))\cdots))\end{aligned}$$

   and, since the branches of T_{n+1} are generic for the appropriate sentences and A is on T_{n+1}, this is equivalent to

$$(\exists \sigma \subseteq A)[\sigma \Vdash^{T_{n+1}} x \in H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X))\cdots))].$$

   By definability of local forcing XII.1.26, $J(A)$ is then r.e. in $A \oplus \emptyset^{(\omega)}$. Similarly, the complement of $J(A)$ is r.e. in $A \oplus \emptyset^{(\omega)}$ because $\langle n, x\rangle \notin J(A)$ is equivalent to

$$(\exists \sigma \subseteq A)[\sigma \Vdash^{T_{n+1}} \neg(x \notin H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X))\cdots)))].$$

   Then $J(A)$ is recursive in $A \oplus \emptyset^{(\omega)}$.

2. $A \oplus \emptyset^{(\omega)} \leq_T C$
   By definition of A, since $A \leq_T T \oplus C$ and $T \leq_T \emptyset^{(\omega)} \leq_T C$.

3. $C \leq_T J(A)$
   Suppose we know $J(A)$, and in particular A (which is its first column). To know C, it is enough to determine for which σ does $T(\sigma) \subseteq A$ hold, because $A = \bigcup_{\sigma \subseteq C} T(\sigma)$.

   Since the first n levels of T are the same as the first n levels of T_n, it is enough to inductively determine the string at the n-th level of T_n contained in A, for each n.

   Obviously, $T_0(\emptyset) = \emptyset \subseteq A$. We now look at $T_1(0)$ and $T_1(1)$. Since we know A, we also know for which $i = 0, 1$ does $T_0(i) \subseteq A$ hold. The only problem is to know whether $T_1(i)$ is equal to $T_0(i)$, and this depends on whether $T_1(i)$ forces $0 \in H_{f(0)}(X)$ or not. In the positive case, $T_1(i)$ is a string extending $T_0(i)$ and forcing that sentence on T_0. In the negative case, $T_1(i) = T_0(i)$ because there is no such string extending $T_0(i)$, and hence $T_0(i)$ forces the negation of that sentence on T_0. But we know $J(A)$, and in particular we know whether $0 \in H_{f(0)}(A)$. Moreover, A is generic for all the appropriate sentences, and thus we also know whether

   $$A \Vdash^{T_0} 0 \in H_{f(0)}(X).$$

   If this is not the case, then we determined that $T_1(i) = T_0(i)$. Otherwise, we still have to find the smallest string $\alpha \supseteq T_0(i)$ such that

   $$\alpha \Vdash^{T_0} 0 \in H_{f(0)}(X),$$

   which can be found recursively once we know that it exists, because $H_{f(0)}$ is an r.e. operator, and thus forcing for such sentences is r.e.

   Knowing whether $T_1(i) \subseteq A$, we determine for which $j = 0, 1$ does $T_2(ij) \subseteq A$ hold in a similar way, and so on. The procedure is obviously general, and it thus permits us to determine inductively for which σ does $T(\sigma) \subseteq A$ hold. $\square$

Note that the same strategy used above is already required for finite iterations of hops, because they are not degree-invariant, and cannot simply be relativized. Relativization works instead for finite iterations of jump.

The rest of the work done in Section XI.1 for hops, which produced a number of interesting consequences for the jump operator, is reproduced in Section XIII.4 for ω-hops, with similar consequences for the ω-jump operator.

In the different direction pursued in the next section, XII.2.22 is used to derive results about minimal covers (XII.3.12), and improved to produce a definition of the set $\mathcal{A}$ of Turing degrees of arithmetical sets (XII.3.14).

XII.3 Turing Degrees of Arithmetical Sets

In this section we consider the class $\boldsymbol{\mathcal{A}}$ of Turing degrees of arithmetical sets. First we briefly look at its local and global properties, following as usual the path set up in Chapter V. Then we turn to the relationships with the global structure of degrees $\boldsymbol{\mathcal{D}}$. On the one hand, we exhibit explicit elementary differences. On the other hand, we show how to define $\boldsymbol{\mathcal{A}}$ in $\boldsymbol{\mathcal{D}}$, thus finally settling a matter left open in Chapter V (see V.7.9).

Local and global properties

Recall that

$$\boldsymbol{\mathcal{A}} = \{\boldsymbol{a} : (\exists n)(\boldsymbol{a} \le \mathbf{0}^{(n)})\}.$$

The next exercises review some of the main local properties discussed in Sections 2 to 6 of Chapter V.

Exercises XII.3.1 a) *For any $m \ge 0$ and $n > 1$, there are degrees between $\mathbf{0}^{(m)}$ and $\mathbf{0}^{(m+n)}$ incomparable with every $\mathbf{0}^{(m+i)}$ with $0 < i < n$. Moreover, every countable partial ordering is embeddable in such degrees.* (Kleene and Post [1954]) (Hint: for $m = 0$, build as in V.2.7 a countable recursively independent set of degrees whose join forms a minimal pair with $\mathbf{0}^{(n-1)}$. For $m > 0$, relativize to $\mathbf{0}^{(m)}$.)

b) *The range of the jump operator restricted to $\boldsymbol{\mathcal{A}}$ is the cone of $\boldsymbol{\mathcal{A}}$ above $\mathbf{0}'$.* (Hint: from V.2.24, since $\boldsymbol{\mathcal{A}}$ is a jump-ideal.)

c) *Every arithmetical ideal of $\boldsymbol{\mathcal{A}}$ has an exact pair in $\boldsymbol{\mathcal{A}}$.* (Spector [1956]) (Hint: from V.4.3, since an exact pair for $\{B_n\}_{n\in\omega}$ can be found recursively in $(\oplus_{n\in\omega} B_n)'$.)

d) *Not every ideal of $\boldsymbol{\mathcal{A}}$ has an exact pair in $\boldsymbol{\mathcal{A}}$.* (Hint: see V.4.6.)

e) *For every n, there are $\Delta^0_{n+2} - \Sigma^0_{n+1}$ sets of minimal degree.* (Manaster [1971]) (Hint: for $n = 0$, see XI.4.5 and X.2.8. For $n > 0$, build a minimal degree as in V.5.11, by diagonalizing at stage $2e + 1$ against the e-th Σ^0_{n+1} set. The diagonalization step is then recursive in $\emptyset^{(n+1)}$, while the minimality step is recursive in $\emptyset''$.)

f) *For every $n > 0$, there are $\Sigma^0_{n+2} - \Delta^0_{n+2}$ sets of minimal degree.* (Jockusch) (Hint: build a strongly uniform tree of minimal degrees recursive in $\emptyset''$, as in V.6.10.b. Choose A and B on the tree as follows. At level n, let x be the unique element on which the branches differ. Let A follow the branch that on x agrees with $\emptyset^{(n+2)}$, and let B follow the other branch. Since $n > 0$, A is r.e. in $\emptyset^{(n+1)}$, and hence Σ^0_{n+2}. Moreover, $\emptyset^{(n+2)} \le_T A \oplus B$ because to know whether $n \in \emptyset^{(n+2)}$ one only has to look at the value of A on the n-th element at which A and B differ. Then A cannot be recursive in $\emptyset^{(n+1)}$, otherwise B would also be, and hence $\emptyset^{(n+2)}$.)

As for the only case left open by part f), Cooper has proved that *there are $\Sigma^0_2 - \Delta^0_2$ sets of minimal degree.*

We now turn to initial segments. On the one hand, the relativization of XI.3.16 shows that $\boldsymbol{\mathcal{D}}(\le \boldsymbol{a})$ must be $\boldsymbol{a}^{(4)}$-presentable. On the other hand, the usual embedding techniques (see Section V.6, Epstein [1979] and Lerman

[1983]) produce an initial segment isomorphic to a given countable uppersemilattice with least element, recursively in two jumps of the presentation. In particular, *the possible isomorphism types of principal ideals of $\boldsymbol{\mathcal{A}}$ are exactly the arithmetical uppersemilattices with least and greatest element*. It follows (from V.6.16.b and as on p. 708) that *the ordinals isomorphic to initial segments of $\boldsymbol{\mathcal{A}}$ are exactly the $\alpha \leq \omega_1^{ck}$*.

We turn now to global properties of $\boldsymbol{\mathcal{A}}$, along the lines of Section V.7. Most of the work has already been done in V.7.6, and the proofs of the results below easily follow from this. Alternative proofs can be given by relying on the stronger (but more difficult) work done for XI.5.4.

Theorem XII.3.2 Complexity of the theory of $\boldsymbol{\mathcal{A}}$ (Nerode and Shore [1980]) *The first-order theory of $\boldsymbol{\mathcal{A}}$ has the same degree (and actually the same isomorphism type) as the theory of Second-Order Arithmetic with set quantifiers restricted to arithmetical sets. In particular, it has degree $\mathbf{0}^{(\omega+\omega)}$.*

Proof. The first statement follows from V.7.6 applied to the jump-ideal $\boldsymbol{\mathcal{A}}$. The second statement follows from this by relativization of the fact that the theory of First-Order Arithmetic has degree $\emptyset^{(\omega)}$. □

Corollary XII.3.3 (Jockusch [1973]) *$\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{A}}$ are not elementarily equivalent.*

Proof. By V.7.3 the first-order theory of $\boldsymbol{\mathcal{D}}$ has the same degree as the theory of Second-Order Arithmetic, in particular degree greater than $\mathbf{0}^{(\omega+\omega)}$. □

The proof of the corollary is indirect, since it only compares the complexities of the two theories. In the next subsection, we produce *an explicit difference between $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{A}}$, at the four-quantifier level*. It is not known whether the theories of $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{A}}$ differ at the three-quantifier level. *There can be no difference at the two-quantifier level*, because V.4.13 and the embeddability of every finite uppersemilattice with least element as an initial segment of $\boldsymbol{\mathcal{A}}$ imply, as on p. I.492, that *the two-quantifier theory of $\boldsymbol{\mathcal{A}}$ is decidable*, with the same decision procedure as that for $\boldsymbol{\mathcal{D}}$. Finally, as for $\boldsymbol{\mathcal{D}}$ and with the same proof, *the three-quantifier theory of $\boldsymbol{\mathcal{A}}$ is undecidable* (Schmerl, see Lerman [1983]).

We turn now to homogeneity questions. The obvious way of relativizing $\boldsymbol{\mathcal{A}}$ to a degree $\boldsymbol{a}$ is by considering the set $\boldsymbol{\mathcal{A}}^{\boldsymbol{a}}$ of degrees above $\boldsymbol{a}$ and arithmetical in it, i.e. below $\boldsymbol{a}^{(\boldsymbol{n})}$ for some n. The next result is proved as in (and is actually a weaker version of) V.7.13.

Theorem XII.3.4 Failure of homogeneity for $\boldsymbol{\mathcal{A}}$ (Nerode and Shore [1980a]) *If $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}}^{\boldsymbol{a}}$ are elementarily equivalent, then $\boldsymbol{a}$ is arithmetical.*

Proof. If $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}}^{\boldsymbol{a}}$ are elementarily equivalent then, by V.7.6, so are the theories of Second-Order Arithmetic with set quantifiers restricted, respectively, to sets arithmetical and arithmetical in $\boldsymbol{a}$. Then $\boldsymbol{a}$ is arithmetical, otherwise the sentence saying that there is a nonarithmetical set would distinguish them (being false in the former and true in the latter). $\square$

Exercise XII.3.5 *The result does not relativize.* (Hint: as in V.7.19.b, using the fact that the set of degrees $\boldsymbol{a}$ such that $\boldsymbol{\mathcal{A}}^{\boldsymbol{a}}$ satisfies a given formula φ is a hyperarithmetical set, because set quantification over sets arithmetical in a fixed set A can be replaced by number quantification over indices relative to $A^{(\omega)}$. Then only Borel Determinacy is needed in V.1.16 to show that either the given set of degrees or its complement contains a cone.)

As usual, to obtain a result that fully relativizes we need to look at isomorphism, rather than elementary equivalence.

Theorem XII.3.6 Failure of strong homogeneity for $\boldsymbol{\mathcal{A}}$. *If $\boldsymbol{\mathcal{A}}^{\boldsymbol{a}}$ and $\boldsymbol{\mathcal{A}}^{\boldsymbol{b}}$ are isomorphic, then $\boldsymbol{a}$ and $\boldsymbol{b}$ are arithmetically equivalent.*

Proof. By V.7.7, two isomorphic jump-ideals are identical. Thus if $\boldsymbol{\mathcal{A}}^{\boldsymbol{a}}$ and $\boldsymbol{\mathcal{A}}^{\boldsymbol{b}}$ are isomorphic, then $\boldsymbol{a} \in \boldsymbol{\mathcal{A}}^{\boldsymbol{b}}$ and $\boldsymbol{b} \in \boldsymbol{\mathcal{A}}^{\boldsymbol{a}}$. In particular, $\boldsymbol{a}$ and $\boldsymbol{b}$ are each arithmetical in the other. $\square$

Finally, we turn to automorphisms of $\boldsymbol{\mathcal{A}}$. The previous results for $\boldsymbol{\mathcal{A}}$ were all obtained by playing with the original proofs of the corresponding results for $\boldsymbol{\mathcal{D}}$, but for the next result a genuinely new proof is required, since the proof of V.7.22 worked explicitly with degrees above all the arithmetical ones.

Theorem XII.3.7 Restriction on automorphisms of $\boldsymbol{\mathcal{A}}$ (Nerode and Shore [1980a]) *Every automorphism of $\boldsymbol{\mathcal{A}}$ is the identity on a cone of $\boldsymbol{\mathcal{A}}$.*

Proof. By the proof of V.7.6, there are finite sequences of arithmetical degrees $\vec{\boldsymbol{a}}$ and $\vec{\boldsymbol{c}}$ coding a standard model of arithmetic of uniformly low degree, and $\vec{\boldsymbol{d}}$ coding the graph of a function g enumerating the natural numbers $\boldsymbol{i_n}$ of the model. Fix a degree $\boldsymbol{b}$ above all of $\mathbf{0}'$, $\vec{\boldsymbol{a}}$, $\vec{\boldsymbol{c}}$ and $\vec{\boldsymbol{d}}$. Consider any degree $\boldsymbol{x} \geq \boldsymbol{b}$, a set X of that degree, and the set

$$\mathcal{X} = \{\boldsymbol{i_n} : n \in X\}.$$

Since $\boldsymbol{x}$ is above $\vec{\boldsymbol{d}}$, $\mathcal{X}$ has degree $\boldsymbol{x}$. Since the degrees coding the model are uniformly low, any finite join of them (and hence of degrees in $\mathcal{X}$) has a jump recursive in $\emptyset'$, and thus an examination of the proof of V.7.2 shows that $\mathcal{X}$ can be coded by parameters $\vec{\boldsymbol{z}}$ below $\boldsymbol{x}$, because $\boldsymbol{b}$ is above $\mathbf{0}'$ (the main point here being that parameters for a set $\{C_n\}_{n\in\omega}$ can be obtained recursively in $\oplus_{n\in\omega}(\oplus_{m\leq n}C_m)'$, and for the set $\mathcal{X}$ this expression is recursive in X, as just noted).

Consider now the effect of an automorphism f of $\boldsymbol{\mathcal{A}}$ on all these degrees: $f(\vec{\boldsymbol{a}})$ and $f(\vec{\boldsymbol{c}})$ still code a standard model of arithmetic, and $f(\vec{\boldsymbol{d}})$ still code a function that enumerates the natural numbers of this model. Moreover, $f(\vec{\boldsymbol{z}})$ are below $f(\boldsymbol{x})$, and still code the same set $\mathcal{X}$. The proof of V.7.2 shows that the decoding procedure of a set from parameters defining it is arithmetical in the parameters and uniform in them, and thus there is a fixed n such that a set coded by parameters is recursive in the n-th jump of any degree bounding the parameters. In particular, $\boldsymbol{x} \leq \boldsymbol{(f(x))^{(n)}}$. For later use notice that, in general, $\boldsymbol{z} \leq \boldsymbol{(f(z))^{(n)}}$ whenever $\boldsymbol{z}$ is above $\boldsymbol{b}$.

We now show that, if $\boldsymbol{x}$ is sufficiently large, then $\boldsymbol{x} \leq \boldsymbol{f(x)}$. By the Jump Inversion Theorem relativized to $f(\boldsymbol{b})$ and iterated n times, if $f(\boldsymbol{x})$ is above $(f(\boldsymbol{b}))^{(n)}$, then there is $\boldsymbol{y}$ above $f(\boldsymbol{b})$ such that

$$f(\boldsymbol{x}) \;=\; (f(\boldsymbol{b}))^{(n)} \cup \boldsymbol{y} \;=\; \boldsymbol{y}^{(n)}.$$

Then

$$\begin{aligned} \boldsymbol{x} &= f^{-1}(f(\boldsymbol{b})^{(n)}) \cup f^{-1}(\boldsymbol{y}) \\ &\leq f^{-1}(f(\boldsymbol{b})^{(n)}) \cup \boldsymbol{y}^{(n)} \\ &= f^{-1}(f(\boldsymbol{b})^{(n)}) \cup f(\boldsymbol{x}), \end{aligned}$$

the first equality by applying f^{-1}, the last one because $f(\boldsymbol{x}) = \boldsymbol{y}^{(n)}$, and the inequality because $f^{-1}(\boldsymbol{y}) \leq \boldsymbol{y}^{(n)}$. Indeed, $\boldsymbol{y}$ is above $f(\boldsymbol{b})$, hence $f^{-1}(\boldsymbol{y})$ is above $\boldsymbol{b}$ and, as noted above,

$$f^{-1}(\boldsymbol{y}) \;\leq\; (ff^{-1}(\boldsymbol{y}))^{(n)} = \boldsymbol{y}^{(n)}.$$

Then $\boldsymbol{x}$ will be bounded by $f(\boldsymbol{x})$ if $f^{-1}(f(\boldsymbol{b})^{(n)})$ is, i.e. if $f(\boldsymbol{x})$ (and hence $\boldsymbol{x}$) is sufficiently large.

A symmetric argument (replacing f and f^{-1}) shows that if $\boldsymbol{x}$ is sufficiently large, then $f(\boldsymbol{x})$ will also be bounded by $\boldsymbol{x}$, and thus $\boldsymbol{x} = f(\boldsymbol{x})$. $\square$

The same proof, together with the fact that the set $\boldsymbol{\mathcal{A}}$ is definable in $\boldsymbol{\mathcal{D}}$ (XII.3.14), and hence fixed under every automorphism, shows that *any automorphism of* $\boldsymbol{\mathcal{D}}$ *is the identity on a cone having an arithmetical degree as a base* (Nerode and Shore [1980a], Harrington and Shore [1981], Shore [1981], Jockusch and Shore [1984]), a result slightly better than V.7.22, and quoted after its proof.

It is known that ($\boldsymbol{\mathcal{R}}$, and hence) $\boldsymbol{\mathcal{A}}$ *is an automorphism base for* $\boldsymbol{\mathcal{D}}$ (Slaman and Woodin [199?]). Together with the definability of $\boldsymbol{\mathcal{A}}$ provided by XII.3.14, this shows that *every nontrivial automorphism of $\boldsymbol{\mathcal{D}}$ induces a nontrivial one of $\boldsymbol{\mathcal{A}}$.* In particular, it follows from Cooper [1997] (and Slaman and Woodin [199?]) that *there are exactly countably many automorphisms of $\boldsymbol{\mathcal{A}}$.*

Exercises XII.3.8 (Odifreddi and Shore [1991]) a) *Every elementary map from $\boldsymbol{\mathcal{A}}$ to $\boldsymbol{\mathcal{A}}$ is an automorphism.* (Hint: modify XII.3.7 as in V.7.21.)

b) *Every elementary substructure of $\boldsymbol{\mathcal{A}}$ cofinal in it is equal to $\boldsymbol{\mathcal{A}}$.*

Cones of minimal covers

A direct comparison of the complexities of the theories of $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{A}}$ shows that they are not elementarily equivalent (XII.3.3), but it does not exhibit an explicit elementary difference between them. This can be done by looking at minimal covers (defined in V.5.16).

The next proposition is a generalization of the fact that r.e. degrees (and $\mathbf{0}'$ among them) are not minimal covers, by relativization of X.2.8.

Proposition XII.3.9 (Jockusch and Soare [1970]) *For any n, $\mathbf{0}^{(n)}$ is not a minimal cover.*

Proof. Since we already know that $\mathbf{0}'$ is not a minimal cover and $\mathbf{0}^{(n+1)}$ is r.e. in $\mathbf{0}^{(n)}$, the obvious attempt is to proceed by induction on n. Suppose that $\mathbf{0}^{(n)}$ is not a minimal cover, and let $\boldsymbol{a}$ be below $\mathbf{0}^{(n+1)}$. Consider $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})}$, which is between $\boldsymbol{a}$ and $\mathbf{0}^{(n+1)}$. There are three cases:

- $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})}$ is strictly between $\boldsymbol{a}$ and $\mathbf{0}^{(n+1)}$. Then $\mathbf{0}^{(n+1)}$ is not a minimal cover of $\boldsymbol{a}$.
- $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})} = \boldsymbol{a}$. Then $\boldsymbol{a} \geq \mathbf{0}^{(\boldsymbol{n})}$, and since $\mathbf{0}^{(n+1)}$ is r.e. in $\mathbf{0}^{(n)}$, it is also r.e. in $\boldsymbol{a}$, and hence not a minimal cover of $\boldsymbol{a}$.
- $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})} = \boldsymbol{a} \cup \mathbf{0}^{(n+1)}$. We would like to use the induction hypothesis on $\mathbf{0}^{(n)}$. However, the hypothesis applies only to degrees $\boldsymbol{a}$ below $\mathbf{0}^{(n)}$, hence certainly not in the present case.

The obvious way out is to strengthen the induction hypothesis to take care of the only troublesome case. That is, we prove not only that $\mathbf{0}^{(n)}$ is not a minimal cover, but that $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})}$ *is not a minimal cover of* $\boldsymbol{a}$, *for every degree* $\boldsymbol{a}$.

We proceed again by induction, with nothing to prove for $n = 0$. For $n+1$, consider $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})}$, which is between $\boldsymbol{a}$ and $\boldsymbol{a} \cup \mathbf{0}^{(n+1)}$. There are three cases:

- $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})}$ is strictly between $\boldsymbol{a}$ and $\boldsymbol{a} \cup \mathbf{0}^{(n+1)}$. Then $\boldsymbol{a} \cup \mathbf{0}^{(n+1)}$ is not a minimal cover of $\boldsymbol{a}$.

- $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})} = \boldsymbol{a}$. Then $\boldsymbol{a} \geq \mathbf{0}^{(\boldsymbol{n})}$, and since $\mathbf{0}^{(n+1)}$ is r.e. in $\mathbf{0}^{(n)}$, $\boldsymbol{a} \cup \mathbf{0}^{(n+1)}$ is r.e. in $\boldsymbol{a}$, and hence not a minimal cover of $\boldsymbol{a}$.
- $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})} = \boldsymbol{a} \cup \mathbf{0}^{(n+1)}$. By the induction hypothesis on n, this is not a minimal cover of $\boldsymbol{a}$. $\Box$

Corollary XII.3.10 *In $\boldsymbol{\mathcal{A}}$ there is no cone of minimal covers.*

Proof. Any degree in $\boldsymbol{\mathcal{A}}$ is bounded by some $\mathbf{0}^{(n)}$. $\Box$

Since we already know from V.5.17 that in $\boldsymbol{\mathcal{D}}$ there are cones of minimal covers, the Σ^0_4 sentence

$$\begin{aligned} \varphi &\Leftrightarrow \text{there is a cone of minimal covers} \\ &\Leftrightarrow (\exists \boldsymbol{a})(\forall \boldsymbol{b} \geq \boldsymbol{a})(\exists \boldsymbol{c} < \boldsymbol{b})(\forall \boldsymbol{d} \leq \boldsymbol{b}) \neg (\boldsymbol{c} < \boldsymbol{d} < \boldsymbol{b}) \end{aligned}$$

is true in $\boldsymbol{\mathcal{D}}$ but not in $\boldsymbol{\mathcal{A}}$, and it provides an explicit elementary difference between the two theories.

However, the proof of V.5.17 is unsatisfactory for three reasons: it is set-theoretical; it uses Arithmetical Determinacy, which we have not yet proved (see Volume III); and it does not exhibit an explicit base for the cone of minimal covers in $\boldsymbol{\mathcal{D}}$. The next corollary overcomes these defects.

Definition XII.3.11 *M mapping sets to sets is a* **minimal cover operator** *if, for every X, the degree of $M(X)$ is a minimal cover of the degree of X.*

Theorem XII.3.12 (Jockusch and Shore [1984]) *There is a minimal cover operator that is an ω-hop.*

Proof. We obviously consider the construction of a minimal degree, which produces a minimal cover of X when relativized to X. The problem is to choose the appropriate construction of minimal degree, i.e. one that (when relativized) produces an ω-hop, up to degree.

1. *There is a minimal cover operator M such that, for some recursive function g and every X, $M(X)$ is the limit of a 0,1-valued function uniformly recursive in X that changes at most $g(x)$ times on x*
   Consider the full approximation construction of a set $A \leq_T \mathcal{K}$ of minimal degree. With notation as in XI.4.8, $A = \lim_{s\to\infty} T^s_s(\emptyset)$. By construction, $T^s_s(\emptyset)$ can change only to reach a higher s-state, hence at most 2^s times. Since $|T^s_s(\emptyset)| \geq s$, it follows that if

$$f(x,s) = \begin{cases} T^s_s(\emptyset)(x) & \text{if } x \leq |T^s_s(\emptyset)| \\ 0 & \text{otherwise} \end{cases}$$

then $A(x) = \lim_{s\to\infty} f(x,s)$, and $f(x,s)$ can change at most $g(x) = 2^x$ times.

The general result is obtained by relativizing to any set X, and by noting that while the function f becomes recursive in X (because relativization changes T_0 from the identity tree to a recursively pointed tree of the same degree as X), the bound on the number of changes of $T_s^s(\emptyset)$ still depends, as before, only on s.

2. *There is a minimal cover operator M that is an ω-hop*
   Now let $A(x) = \lim_{s\to\infty} f(x,s)$, with $f(x,s)$ a 0,1-valued recursive function that can change at most $g(x)$ times. We simply generalize the argument of XI.5.10.a. Let

$$\begin{aligned}\langle x,s\rangle \in A_n \quad \Leftrightarrow \quad & f(x,s) \text{ has changed } g(x)-n \text{ times before } s,\\ & \text{and it is not correct.}\end{aligned}$$

We want to show that A_0 is r.e., A_n is recursive in A_{n+1}, and A_{n+1} is r.e. in A_n, uniformly in n, and $A \equiv_T \oplus_{n\in\omega} A_n$. Then A has the same degree as the ω-hop defined by the A_n.

A_0 is r.e. because it is empty. Indeed, if $f(x,s)$ has changed $g(x)$ times, then it cannot change any more, and it is correct.

$A_n \leq_T A_{n+1}$ because $f(x,s)$ is not correct if and only if the values different from it are correct, since f only takes values 0 and 1. Given s, if t_s is the greatest $t < s$ such that $f(x,t) \neq f(x,s)$, then

$$\begin{aligned}\langle x,s\rangle \in A_n \quad \Leftrightarrow \quad & f(x,s) \text{ has changed } g(x)-n \text{ times before } s,\\ & \text{and } \langle x,t_s\rangle \notin A_{n+1},\end{aligned}$$

and so $A_n \leq_T A_{n+1}$, uniformly in n.

A_{n+1} is r.e. in A_n because, when $f(x,s)$ has changed $g(x)-n-1$ times, three things may happen: either $f(x,s)$ does not change any more (then it is correct, and $\langle x,s\rangle \notin A_{n+1}$); or it changes once more, at a stage $t > s$, and then $\langle x,t\rangle$ can be in A_n or not. If $\langle x,t\rangle \in A_n$, then the value of $f(x,t)$ is not correct by definition of A_n, and the previous value $f(x,s)$ is correct (since f is 0,1-valued); then $\langle x,s\rangle \notin A_{n+1}$. If $\langle x,t\rangle \notin A_n$, then, since f has changed $g(x)-n$ times before stage t, $f(x,t)$ is correct and thus $f(x,s)$ is not; then $\langle x,s\rangle \in A_{n+1}$. Thus

$$\begin{aligned}\langle x,s\rangle \in A_{n+1} \quad \Leftrightarrow \quad & f(x,s) \text{ has changed } g(x)-n-1 \text{ times before } s,\\ & \text{and } (\exists t > s)(\langle x,t\rangle \notin A_n),\end{aligned}$$

and so A_{n+1} is r.e. in A_n, uniformly in n.

Finally, $(\oplus_{n\in\omega} A_n) \leq_T A$ because by definition

$$\begin{aligned} \langle x, s\rangle \in A_n \quad \Leftrightarrow \quad & f(x,s) \text{ has changed } g(x)-n \text{ times before } s, \\ & \text{and } f(x,s) \neq A(x), \end{aligned}$$

and $A \leq_T \oplus_{n\in\omega} A_n$ because, by supposing $f(x,0)=0$ for every x (which can always be done),

$$x \in A \;\Leftrightarrow\; \langle x, 0\rangle \in A_{g(x)}.$$

Indeed, $f(x,0)$ has changed 0 times at stage 0. And $x \in A$ if and only if 0 is not correct as an approximation of $A(x)$. Thus $A \equiv_T \oplus_{n\in\omega} A_n$.

Having A, by relativization of its construction to X we obtain the desired ω-hop. $\square$

Corollary XII.3.13 (Jockusch [1973], Harrington and Kechris [1975], Jockusch and Shore [1984], [1985]) $\mathbf{0}^{(\omega)}$ *is the base of a cone of minimal covers.*

Proof. By the ω-Hops Inversion Theorem XII.2.22, every degree $\boldsymbol{c}$ above $\mathbf{0}^{(\omega)}$ is the degree of $M(A)$ for some A, and hence is a minimal cover (of the degree of A). $\square$

Definability of the Turing degrees of arithmetical sets

Many absolute results on definability, homogeneity and automorphisms for $\boldsymbol{\mathcal{D}}$ were proved in Section V.7 by relying on the definability of $\boldsymbol{\mathcal{A}}$, which we can now finally prove.

The results of the last subsection provide the necessary hints. On the one hand, since there is no cone of minimal covers in $\boldsymbol{\mathcal{A}}$ but $\mathbf{0}^{(\omega)}$ is the base of such a cone, the definable set

$$\boldsymbol{\mathcal{B}} \;=\; \{\boldsymbol{b} : \boldsymbol{b} \text{ is not the base of a cone of minimal covers}\}$$

separates $\boldsymbol{\mathcal{A}}$ and the cone above $\mathbf{0}^{(\omega)}$ (however, $\boldsymbol{\mathcal{A}} \neq \boldsymbol{\mathcal{B}}$ because there are nonarithmetical degrees which are not minimal covers; for example, by V.6.16.b there is a nonarithmetical degree that is the top of an initial segment of order type $\omega+1$). On the other hand, it is not known whether every base of a cone of minimal covers is $\geq \mathbf{0}^{(\boldsymbol{\omega})}$.

An examination of the proof of XII.3.9 shows that we proved not only that $\mathbf{0}^{(n)}$ is not a minimal cover, but that for any degree $\boldsymbol{a}$, $\boldsymbol{a} \cup \mathbf{0}^{(\boldsymbol{n})}$ is not a minimal cover of $\boldsymbol{a}$. It is thus natural to consider the definable set $\boldsymbol{\mathcal{C}}$ of degrees with this property (to which all $\mathbf{0}^{(n)}$ belong), and its downward closure (which then contains $\boldsymbol{\mathcal{A}}$). The next result shows that $\boldsymbol{\mathcal{C}}$ coincides with $\boldsymbol{\mathcal{A}}$.

Theorem XII.3.14 Definability of the Turing degrees of arithmetical sets (Harrington and Shore [1981], Jockusch and Shore [1984]) *The set of arithmetical degrees is definable in $\boldsymbol{\mathcal{D}}$. Precisely, $\boldsymbol{\mathcal{A}}$ is the downward closure of*

$$\boldsymbol{\mathcal{C}} = \{\boldsymbol{b} : (\forall \boldsymbol{a})(\boldsymbol{a} \cup \boldsymbol{b} \textit{ is not a minimal cover of } \boldsymbol{a})\}.$$

Proof. We have already argued that, by the proof of XII.3.9, the downward closure of $\boldsymbol{\mathcal{C}}$ contains $\boldsymbol{\mathcal{A}}$.

For the converse, we want to show that if $\boldsymbol{b}$ is not arithmetical then there is a degree $\boldsymbol{a}$ such that $\boldsymbol{a} \cup \boldsymbol{b}$ is a minimal cover of $\boldsymbol{a}$, so that $\boldsymbol{b}$ is not in $\boldsymbol{\mathcal{C}}$. The idea is to obtain

$$\boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{m}(\boldsymbol{a})$$

where $\boldsymbol{m}(\boldsymbol{a})$ is the degree of $M(A)$ for some set A of degree $\boldsymbol{a}$, where M is the ω-hop of XII.3.12. Then $\boldsymbol{a} \cup \boldsymbol{b}$ is a minimal cover of $\boldsymbol{a}$ because so is $\boldsymbol{m}(\boldsymbol{a})$.

From the ω-Hops Inversion Theorem XII.2.22, we know how to obtain a degree $\boldsymbol{a}$ such that

$$\boldsymbol{m}(\boldsymbol{a}) = \boldsymbol{c}$$

for any given degree $\boldsymbol{c}$ above $\mathbf{0}^{(\omega)}$. Moreover, if $\boldsymbol{c}$ is also above $\boldsymbol{b}$, then from (the extension of) the Cupping Theorem (quoted after) XI.3.10 we know how to obtain a degree $\boldsymbol{a}$ such that

$$\boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{c}.$$

Thus consider any $\boldsymbol{c} \geq \boldsymbol{b} \cup \mathbf{0}^{(\boldsymbol{\omega})}$, which is above both $\mathbf{0}^{(\omega)}$ and $\boldsymbol{b}$. We have to show that it is possible to combine XII.2.22 and XI.3.10 to produce a degree $\boldsymbol{a}$ such that, simultaneously,

$$\boldsymbol{m}(\boldsymbol{a}) = \boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{c}.$$

We are thus reduced to proving the following result: *given a nonarithmetical set B and a set C such that $B \oplus \emptyset^{(\omega)} \leq_T C$, there is a set A such that*

$$M(A) \equiv_T A \oplus B \equiv_T C.$$

The idea is to modify the proof of XI.3.10, by changing the forcing used there to make A 1-generic, into the local forcing used in XII.2.22 to control the ω-hop M.

To be able to apply the methods of XII.2.22 directly, we need a special representation of M as an ω-hop (since the result is not known to hold in general). The particular representation obtained in XII.3.12 has the property that there is a 0,1-valued function f uniformly recursive in X that approximates

$M(X)$ in the limit, such that $f(x,0)=0$ and $f(x,s)$ can change at most $g(x)$ times, for a fixed recursive function g. One then considers the sequence of sets

$$\begin{aligned} \langle x,s\rangle \in A_n \quad \Leftrightarrow \quad & f(x,s) \text{ has changed } g(x)-n \text{ times before } s, \\ & \text{and it is not correct.} \end{aligned}$$

Since we will work with one fixed x at a time, it is convenient to consider not A_n as a set of pairs, but its projection A_n^x on x, i.e.

$$s \in A_n^x \;\Leftrightarrow\; \langle x,s\rangle \in A_n.$$

A_0^x is r.e. in X and A_{n+1}^x is r.e. in $X \oplus A_n^x$ for every n. Thus there is a recursive function h (obtained by the uniformities of the definition of A_n^x) such that

$$X \oplus A_n^x = H_{h(n,x)}(H_{h(n-1,x)}(\cdots(H_{h(0,x)}(X))\cdots)),$$

and h defines an ω-hop Turing equivalent to $M(X)$. Moreover, from the proof of XII.3.12 one also has

$$\begin{aligned} x \in M(X) \quad & \Leftrightarrow \quad \langle x,0\rangle \in A_{g(x)} \\ & \Leftrightarrow \quad 0 \in A_{g(x)}^x \\ & \Leftrightarrow \quad 1 \in X \oplus A_{g(x)}^x \\ & \Leftrightarrow \quad 1 \in H_{h(g(x),x)}(H_{h(g(x)-1,x)}(\cdots(H_{h(0,x)}(X))\cdots)), \end{aligned}$$

where 1 arises from the definition of join, since

$$z \in D \;\Leftrightarrow\; 2z+1 \in C \oplus D.$$

We will thus have to control the first $g(x)$ levels of the ω-hop defined by h, to be able to control the membership of x in $M(X)$.

We now turn to the construction. We define a sequence of trees $\{T_e\}_{e\in\omega}$, with T_e arithmetical and T_{e+1} a subtree of T_e, and we will let $A = \bigcup_{e\in\omega} T_e(\emptyset)$. We will closely follow the proof of XI.3.10. In particular, as there we fix a recursive sequence $\{\tau_n\}_{n\in\omega}$ of mutually incompatible strings, e.g.

$$\tau_n = \underbrace{0\cdots0}_{n \text{ times}}\,1.$$

The place of the string $\sigma_e * \tau_n$ in the proof of XI.3.10 will now be taken by the string $T_e(\tau_n)$.

We start with

$$T_0 \;=\; \text{identity tree.}$$

At stage $e+1$, given T_e we would like to look for the smallest n such that

$$\begin{array}{l} n \notin B \wedge (\forall A \supseteq T_e(\tau_n))(e \in M(A)), \text{ or} \\ n \in B \wedge (\forall A \supseteq T_e(\tau_n))(e \notin M(A)). \end{array}$$

To control the truth-value of $e \in M(A)$, by the particular representation of M given above we need to work on a tree whose branches are generic w.r.t. the sentences of the form

$$x \in H_{h(g(e),e)}(H_{h(g(e)-1,e)}(\cdots(H_{h(0,e)}(X))\cdots)).$$

Consider, for each n, the subtree Q_n of T_e above $T_e(\tau_n)$, obtained by $g(e)$ successive iterations of local forcing, the i-th iteration determining the truth-value of all sentences of the form

$$x \in H_{h(i,e)}(X),$$

as in the proof of XII.2.22. Since the sentence relative to x is forced by construction at level x of the tree, by what we have seen above the truth-value of $e \in M(A)$ is now determined, for any branch A of Q_n, by which branch of level 1 it extends. Since there are four branches at level 1, the truth-value of $e \in M(A)$ it is not independent of the branch A, but if we consider only the branches above a given fixed string at level 1, e.g. $Q_n(00)$, then 1 belongs either to all or to none of them, and the truth-value of $e \in M(A)$ is now independent of the branch.

Consider

$$S = \{n : \text{for all branches } A \text{ of } Q_n \text{ above } Q_n(00),\ e \in M(A)\}.$$

This is an arithmetical set, because Q_n is obtained by a finite iteration of local forcing on an arithmetical tree T_e, and the value of $e \in M(A)$ does not depend on the branch A, and thus needs to be checked only once (say, for the leftmost branch of Q_n above $Q_n(00)$). Since S is arithmetical and B is not, $S \neq B$. Choose n such that

$$n \in B \Leftrightarrow n \notin S,$$

and let

$$T_{e+1} = \text{full subtree of } Q_n \text{ above } Q_n(00 * \langle C(e)\rangle).$$

As in XII.2.22, we can now prove that $M(A) \equiv_T C$, using the fact that the construction is recursive in $C \oplus \emptyset^{(\omega)}$ because every T_e is uniformly arithmetical, by the definability of local forcing.

As in XI.3.10, we can also prove that $A \oplus B \equiv_T C$, using the fact that the construction is recursive in $A \oplus B \oplus \emptyset^{(\omega)}$. $\square$

A different proof of the result above can be obtained from Cooper's Theorem on the definability of the jump operator (XI.5.17): then $\boldsymbol{\mathcal{A}}$ can be naturally defined as the least jump-ideal (see V.8.1). On the other hand, the proof of Cooper's Theorem is very similar in strategy to the one given above, except that it works at level 2 instead of at level ω of the Hop Hierarchy (see p. 726).

æ

Chapter XIII

Arithmetical Degrees

Chapter XII provided a microscopic analysis of the arithmetical sets. We now proceed to study the continuum modulo the arithmetical sets. The first section of this chapter is devoted to a study of the structure $(\boldsymbol{\mathcal{D}_a}, \leq)$ of the **arithmetical degrees**, where $\leq$ is the partial ordering induced by $\leq_a$ (see IV.1.23). We also consider the structure $\boldsymbol{\mathcal{D}'_a}$, where the jump operator is induced by the ω-jump (V.1.10).

The rest of the chapter deals with an analogue of the r.e. sets in the context of arithmetical computations. Note that the arithmetically enumerable sets are not interesting, since they coincide with the arithmetical sets. Our starting point is the analogy between $\mathcal{K}$ and $\emptyset^{(\omega)}$. In Section 2 we introduce the **$\boldsymbol{\omega}$-r.e.a. set**, for which $\emptyset^{(\omega)}$ is a complete set. In Section 3 we solve the analogue of Post's problem, and in Section 4 we show that the ω-r.e.a. sets behave w.r.t. the ω-jump operator as the r.e. sets do w.r.t. the jump operator. Section 5 establishes an elementary difference with the theory of r.e. T-degrees, by proving the Diamond Theorem for ω-r.e.a. a-degrees.

XIII.1 The Theory of Arithmetical Degrees

This first section is devoted to a study of the following structures, defined in Chapter IV:

1. $\boldsymbol{\mathcal{D}_a}$, consisting of the a-degrees (the equivalence classes w.r.t. the equivalence relation $\equiv_a$) with the partial ordering induced by $\leq_a$, where

$$\begin{aligned} A \leq_a B &\Leftrightarrow A \text{ is arithmetical in } B \\ A \equiv_a B &\Leftrightarrow A \leq_a B \wedge B \leq_a A. \end{aligned}$$

2. $\boldsymbol{\mathcal{D}'_a}$, which is $\boldsymbol{\mathcal{D}_a}$ together with the arithmetical jump operator induced by the ω-jump.

The study of these structures will be parallel to that of their analogues $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{D}'}$. In general, proofs of results for T-degrees will be lifted to proofs of similar results for a-degrees by use of appropriate notions of forcing.

The finite extension method

The simplest method of proof we used for the study of Turing degrees was finite extension, which built sets with required computational properties by finite initial segments. The basic fact that made the use of finite extensions viable was that convergent recursive computations from an oracle only use a finite portion of the oracle. This is no longer true for arithmetical computations, which could easily involve the whole oracle. But we have seen in Chapter XII that arithmetical computations from ω-generic sets still involve only finite parts of the oracle. Whenever we succeed in combining the old proofs with the construction of ω-generic sets, we thus extend the results from T-degrees to a-degrees. This is automatic in many cases, since both the classes of ω-generic sets and of the sets built by an application of the finite extension method are comeager, and thus their intersection is nonempty.

We only give two examples (one exploiting automatic properties of ω-generic sets, the other a direct construction), and leave to the reader the task of extending the results obtained by finite extension from T-degrees to a-degrees.

Proposition XIII.1.1 (Feferman [1965]) *Every countable partial ordering is embeddable in the a-degrees.*

Proof. By V.2.9, it is enough to show that if A is ω-generic, then its components are arithmetically independent, i.e. $A^{[n]} \not\leq_a \oplus_{m\neq n} A^{[m]}$. Suppose that, for some n, $A^{[n]} \leq_a \oplus_{m\neq n} A^{[m]}$. Then, for some arithmetical formula ψ,

$$x \in A^{[n]} \ \Leftrightarrow\ \oplus_{m\neq n} A^{[m]} \models \psi(\overline{x}).$$

There is then an arithmetical formula φ such that

$$X \models \varphi \ \Leftrightarrow\ X^{[n]} \text{ is defined from } \oplus_{m\neq n} X^{[m]} \text{ via } \psi \text{ as above.}$$

Since $A \models \varphi$ and A is ω-generic, $\sigma \Vdash \varphi$ for some $\sigma \subseteq A$.

Define B as follows. $B^{[n]}$ and $A^{[n]}$ are equal, except on some element outside the domain of σ, and $B^{[m]} = A^{[m]}$ for $m \neq n$. Since B differs finitely from A, it is still ω-generic. Since $B \supseteq \sigma$, $B \models \varphi$. Since $\oplus_{m\neq n} B^{[m]} = \oplus_{m\neq n} A^{[m]}$, it then follows that $B^{[n]} = A^{[n]}$, which is a contradiction. $\square$

Corollary XIII.1.2 *The one-quantifier theory of $\mathcal{D}_a$ is decidable.*

Proof. As in V.2.10. $\square$

Exercises XIII.1.3 a) *If A is ω-generic, then it is the join of two sets whose a-degrees form a minimal pair.* (Hint: see XI.2.3.)

b) *If (A, B) is an ω-generic pair then the a-degrees of A and B form a minimal pair.* (Hint: see XI.2.3.)

The next result extends V.2.16, and provides an example of a proof using the analogue of the splitting method.

Proposition XIII.1.4 *Every nonarithmetical a-degree is part of a minimal pair.*

Proof. Given a nonarithmetical set B we want A nonarithmetical such that, for every arithmetical formula ψ and φ,

$$(\forall x)[x \in C \Leftrightarrow A \models \psi(\overline{x}) \Leftrightarrow B \models \varphi(\overline{x})] \Rightarrow C \text{ arithmetical.}$$

Let $(\psi_s, \varphi_s)_{s\in\omega}$ and $\{\alpha_s\}_{s\in\omega}$ be recursive enumerations of the pairs of arithmetical formulas with one free variable and the arithmetical sentences, respectively.

Let $A = \bigcup_{s\in\omega} \sigma_s$ where $\sigma_0 = \emptyset$, and σ_{s+1} is inductively defined from σ_s as follows.

- Let $\sigma \supseteq \sigma_s$ be such that

$$\sigma \Vdash \alpha_s \text{ or } \sigma \Vdash \neg\alpha_s.$$

  This makes A ω-generic.

- See if there are $\sigma_0, \sigma_1 \supseteq \sigma$ and x such that

$$\sigma_0 \Vdash \psi_s(\overline{x}) \text{ and } \sigma_1 \Vdash \neg\psi_s(\overline{x}).$$

  If not, let $\sigma_{s+1} = \sigma$. Then, if

$$x \in C \Leftrightarrow A \models \psi_s(\overline{x}),$$

  C is arithmetical. Indeed, to see if $x \in C$, it is enough to search for $\tau \supseteq \sigma_{s+1}$ such that

$$\tau \Vdash \psi_s(\overline{x}) \text{ or } \tau \Vdash \neg\psi_s(\overline{x}),$$

  which exists by quasi-completeness. Then $x \in C$ if and only if $\tau \Vdash \psi_s(\overline{x})$.

If σ_0 and σ_1 exist, let

$$\sigma_{s+1} = \begin{cases} \sigma_1 & \text{if } B \models \varphi_s(\overline{x}) \\ \sigma_0 & \text{otherwise.} \end{cases}$$

Then A and B do not satisfy the hypothesis relative to (ψ_s, φ_s), and there is nothing to prove. $\Box$

Note that the methodological remarks of Section V.3 also hold in the present context. In particular, it is not possible to build minimal a-degrees by a finite forcing of the kind used so far.

The tree method

The next methodology to be treated, after the finite extension method, would be the coinfinite extension method. A small variation of the forcing method is needed here. Suppose, for example, that we try to adapt the proof of Spector's Theorem V.4.3. Since we want to build an ω-generic set which satisfies some additional properties, at stage $s+1$ we will be given a coinfinite condition θ_s and an arithmetical sentence φ, and we want to ensure that the set $A = \bigcup_{s\in\omega} \theta_s$ we are building will satisfy $A \Vdash \varphi$ or $A \Vdash \neg\varphi$. If θ_s is finite, by quasi-completeness we can obtain an extension of it which forces either φ or $\neg\varphi$, and we have no problem. But if θ_s is infinite, we do not know if we can find such a finite string compatible with θ_s.

The technique for avoiding the obstacle is local forcing, and it is illustrated by the next lemma, which is the analogue of the Minimality Lemma V.5.10 (in the generalized form used in V.5.21).

Proposition XIII.1.5 Minimality Lemma for arithmetical reductions (Sacks [1971]) *Given an arithmetical formula φ and a tree T, let A_φ be the set determined by φ from A:*

$$x \in A_\varphi \;\Leftrightarrow\; A \models \varphi(\overline{x}).$$

Then there is a tree $Q \subseteq T$ arithmetical in T, such that one of the following holds:

1. *for every A on Q, $A_\varphi \leq_a Q$*
2. *for every A on Q, $A \leq_a A_\varphi \oplus Q$.*

Proof. The idea is the same as in V.5.9 and V.5.10, i.e. we build Q with either no φ-splitting on it, or as a φ-splitting tree (with the natural notion of splitting). See if

$$(\forall \sigma \in T)(\exists \tau_1, \tau_2 \in T)(\exists x)[\tau_1, \tau_2 \supseteq \sigma \;\wedge\; \tau_1 \Vdash^T \varphi(\overline{x}) \;\wedge\; \tau_2 \Vdash^T \neg\varphi(\overline{x})].$$

- The first case we consider is when this does not happen, i.e. there is $\sigma \in T$ such that, for every $\tau_1, \tau_2 \supseteq \sigma$ on T and every x,

  $$(\forall i \leq 1)[\tau_i \Vdash^T \varphi(\overline{x}) \vee \tau_i \Vdash^T \neg\varphi(\overline{x})] \;\Rightarrow\; [\tau_1 \Vdash^T \varphi(\overline{x}) \Leftrightarrow \tau_2 \Vdash^T \varphi(\overline{x})].$$

  Fix such a σ. We would be tempted to let Q be the full subtree of T above σ, and claim that 1 holds because there are no φ-splittings on Q. The fact is that we can go from forcing on T to truth only for sets sufficiently generic on T. But, since φ is arithmetical, $\varphi \in \Sigma_n^0$ for some n. Let Q be a tree of n-generic sets on T extending σ, which can be taken to be recursive in $T^{(n)}$, and hence arithmetical in T.

  Now for A on Q we have that A_φ is arithmetical in Q. Indeed, to determine if $x \in A_\varphi$, we only have to search for $\tau \in Q$ such that $\tau \supseteq \sigma$ and $\tau \Vdash^Q \varphi(\overline{x})$ or $\tau \Vdash^Q \neg\varphi(\overline{x})$. Such a τ exists by quasi-completeness, and can be found arithmetically in Q (actually, recursively in $Q^{(n)}$). Since there are no φ-splittings on Q, and every branch of Q is n-generic, it follows that

  $$x \in A_\varphi \;\Leftrightarrow\; \tau \Vdash^Q \varphi(\overline{x}).$$

- We now consider the remaining case, i.e. that no such string σ exists. Fix a recursive enumeration $\{\alpha_s\}_{s\in\omega}$ of all the Σ_n^0 sentences. We build a subtree Q of T as follows. Let $Q(\emptyset) = T(\emptyset)$. Given $Q(\sigma)$ on T, choose $\tau_0, \tau_1 \supseteq Q(\sigma)$ on T, and x, such that

  $$\tau_0 \Vdash^T \varphi(\overline{x}) \;\wedge\; \tau_1 \Vdash^T \neg\varphi(\overline{x}),$$

  which exist by case hypothesis. Then let $Q(\sigma * i)$ be an extension $\hat{\tau}_i$ of τ_i such that $\hat{\tau}_i \Vdash^T \alpha_s$ or $\hat{\tau}_i \Vdash^T \neg\alpha_s$, where $s = |\sigma|$, which exists by quasi-completeness. Q is arithmetical in T. By construction, each branch of Q is n-generic on it (since $Q \subseteq T$) and Q is φ-splitting, i.e. for every σ there exists x such that

  $$Q(\sigma * 0) \Vdash^Q \varphi(\overline{x}) \;\wedge\; Q(\sigma * 1) \Vdash^Q \neg\varphi(\overline{x}).$$

  Then for any A on Q we have $A \leq_a A_\varphi \oplus Q$. Indeed, if we have $Q(\sigma) \subseteq A$ and we want to determine for which i is $Q(\sigma * i) \subseteq A$, we let x be as above. Then

  $$\begin{aligned} x \in A_\varphi &\;\Rightarrow\; Q(\sigma * 0) \subseteq A \\ x \notin A_\varphi &\;\Rightarrow\; Q(\sigma * 1) \subseteq A. \end{aligned}$$

  Thus one can determine A arithmetically in A_φ and Q. $\square$

We have carried out the proof in some detail, but again the argument is nothing more than the proof of the corresponding result for T-degrees, mixed with local forcing.

Theorem XIII.1.6 (Sacks [1971]) *There exists a minimal a-degree.*

Proof. We have to build a set A satisfying the following requirements:

- A is different from the e-th arithmetical set
- if φ is the e-th arithmetical formula with one free variable (i.e. the e-th arithmetical reduction), then either A_φ is arithmetical or $A \leq_a A_\varphi$.

We build a sequence of arithmetical trees as follows:

$$\begin{array}{lcl} T_0 & = & \text{identity tree} \\ T_{2e+1} & = & \text{the full subtree of } T_{2e} \text{ above } T_{2e}(i) \\ & & \text{(for an } i \text{ such that } T_{2e}(i) \text{ differs from the } e\text{-th arithmetical set)} \\ T_{2e+2} & = & \text{the } Q \text{ of the Minimality Lemma, for } T = T_{2e+1}. \end{array}$$

Note that, by induction, T_e is arithmetical. Moreover, $A = \bigcup_{n\in\omega} T_n(\emptyset)$ satisfies all the requirements. □

Exercises XIII.1.7 a) *There are* $2^{\aleph_0}$ *minimal a-degrees*. (Hint: as in V.5.12.b.)

b) *A minimal upper bound for the T-degrees of the arithmetical sets is not of minimal a-degree*. (Hint: if $A \leq_T \emptyset^{(n)}$ for every n, then $\emptyset^{(\omega)} \leq_T A''$ by the proof of V.8.4, and thus $\emptyset^{(\omega)} \leq_a A$. But the a-degree of $\emptyset^{(\omega)}$ is not minimal.)

c) *There is a cone of minimal covers in* $\boldsymbol{\mathcal{D}_a}$. (Hint: relativize XIII.1.6, and apply Borel determinacy to the analogue of V.1.16.)

We can also easily generalize the results on minimal upper bounds, by using the analogue of the notion of recursively pointed tree (V.5.18).

Definition XIII.1.8 (Sacks [1971]) *An* **arithmetically pointed tree** *is a tree which is arithmetical in all of its branches.*

Exercises XIII.1.9 a) *If T is arithmetically pointed, $Q \subseteq T$ and $Q \leq_a T$, then Q is also arithmetically pointed, and $Q \equiv_a T$.* (Hint: see V.5.19.)

b) *If T is arithmetically pointed and $T \leq_a A$, then there is some arithmetically pointed tree $Q \subseteq T$ such that $Q \equiv_a A$.* (Hint: see V.5.20.)

Proposition XIII.1.10 *Every countable set of a-degrees has* $2^{\aleph_0}$ *minimal upper bounds.*

Proof. As in V.5.21 and V.5.22. □

Arithmetical jump

The analogue of the jump operator for a-degrees is the ω-jump.

Exercises XIII.1.11 a) *If $A \leq_a B$, then $A^{(\omega)} \leq_1 B^{(\omega)}$.* (Hint: see V.1.11.a.)

b) *The converse fails.* (Simpson [1985]) (Hint: consider an effective enumeration $\{\varphi_n\}_{n\in\omega}$ of the arithmetical sentences, and the natural construction of an ω-generic set A w.r.t. such an enumeration. Then A is not arithmetical, but $A^{(\omega)} \leq_1 \emptyset^{(\omega)}$ because if $\varphi_{f(n)}$ is the arithmetical sentence $\overline{n} \in X^{(\omega)}$, then

$$n \in A^{(\omega)} \Leftrightarrow A \models \varphi_{f(n)} \Leftrightarrow \sigma_{f(n)+1} \Vdash \varphi_{f(n)},$$

and the construction of A is recursive in $\emptyset^{(\omega)}$.)

c) *$A \leq_1 A^{(\omega)}$ and $A^{(\omega)} \not\leq_a A$.*

Proposition XIII.1.12 Arithmetical Jump Inversion Theorem for $\boldsymbol{\mathcal{D}_a}$ (MacIntyre [1977]) *The range of the arithmetical jump operator is the cone* $\boldsymbol{\mathcal{D}_a(\geq \mathbf{0}'_a)}$.

Proof. Suppose $\emptyset^{(\omega)} \leq_a C$. Then $\emptyset^{(\omega)} \leq_T C^{(n)}$ for some n. By XII.2.4 there is A such that $A^{(\omega)} \equiv_T C^{(n)}$ and thus $A^{(\omega)} \equiv_a C$. $\square$.

By the proof, which uses XII.2.4, *every a-degree above $\mathbf{0}'_a$ is actually the arithmetical jump of an a-degree that realizes the least possible arithmetical jump.*

Exercises XIII.1.13 a) *The arithmetical jump operator is not one-one.* (Hint: an ω-generic set A recursive in $\emptyset^{(\omega)}$ has ω-jump $\emptyset^{(\omega)}$.)

b) *The arithmetical jump operator is never one-one on its range.* (Hint: by relativization of a). Alternatively, given $\emptyset^{(\omega)} \leq_a C$, build ω-generic sets A and B such that $A^{(\omega)} = B^{(\omega)} = A \oplus B = C$.)

Global properties

The methods introduced above show how to extend results from Turing degrees to a-degrees, by mixing their proofs with appropriate forcing arguments. We can thus obtain a great deal of knowledge about $\boldsymbol{\mathcal{D}_a}$. In particular, by extending the definability of countable relations from finitely many parameters (V.7.2), we obtain the following as in V.7.3.

Theorem XIII.1.14 The complexity of the theory of $\boldsymbol{\mathcal{D}_a}$ (Nerode and Shore [1980]) *The first-order theory of $\boldsymbol{\mathcal{D}_a}$ has the same degree (and actually the same isomorphism type) as the theory of Second-Order Arithmetic.*

Corollary XIII.1.15 (Harding [1974]) *The first-order theory of $\boldsymbol{\mathcal{D}_a}$ is undecidable and not axiomatizable.*

The original proofs of these results used (VI.4.7 and) the embeddability of every countable distributive lattice as an initial segment of $\boldsymbol{\mathcal{D}_a}$, proved by Harding [1974]. The best known result in this direction is that *a partial ordering with power at most $\aleph_1$ is isomorphic to an initial segment of $\boldsymbol{\mathcal{D}_a}$ if and only if it is an uppersemilattice with a least element, and countable predecessor property* (Simpson [1985]). The proof once again mixes the corresponding proof for Turing degrees, given by Abraham and Shore [1986], with local forcing.

As for the usual global properties, the proofs given for $\boldsymbol{\mathcal{D}}$ almost effortlessly produce similar results for $\boldsymbol{\mathcal{D}'_a}$. We just provide some samples.

Theorem XIII.1.16 Absolute definability for $\boldsymbol{\mathcal{D}'_a}$ (Shore) *A relation on a-degrees above $\mathbf{0}'_{\boldsymbol{a}}$ is definable in $\boldsymbol{\mathcal{D}'_a}$ if and only if it is definable in Second-Order Arithmetic.*

Proof. We indicate the changes needed in the proof of V.7.10 to show that, for every standard model of arithmetic in the a-degrees below $\mathbf{0}'_{\boldsymbol{a}}$, the map taking an a-degree $\boldsymbol{x}$ above $\mathbf{0}'_{\boldsymbol{a}}$ into a set (of natural numbers in the standard model) of a-degree $\boldsymbol{x}$ is definable in $\boldsymbol{\mathcal{D}'_a}$, with parameters the a-degrees coding the standard model. Then the same proof of V.7.11 produces the required result.

The main difference is that, given an a-degree $\boldsymbol{z}$ and parameters below it coding a set, we can recover this set recursively in $\boldsymbol{z} \cup \mathbf{0}'_{\boldsymbol{a}}$. The reason is that we also need to recover the standard part of the model, and this involves a few quantifiers over the parameters coding it: now, these parameters are below $\mathbf{0}'_{\boldsymbol{a}}$ by hypothesis, and the recovering procedure is arithmetical in them, hence everything takes place in the a-degrees below $\mathbf{0}'_{\boldsymbol{a}}$.

Then, given a set X (of natural numbers in the standard model) of a-degree $\boldsymbol{x}$, we wish to prove that

$$\boldsymbol{x} = deg_a(X) \;\Leftrightarrow\; \boldsymbol{x} \text{ is the l.u.b. of } \boldsymbol{\mathcal{X}},$$

where $deg_a(X)$ is the a-degree of X, and

$$\mathcal{X} = \{\boldsymbol{z} : \text{the sets coded by parameters below } \boldsymbol{z} \cup \mathbf{0}'_{\boldsymbol{a}} \text{ are arithmetical in } X\}.$$

Being definable in Second-Order Arithmetic with parameters $\mathbf{0}'_{\boldsymbol{a}}$ and the a-degrees coding the standard model, $\mathcal{X}$ is definable in $\boldsymbol{\mathcal{D}_a}$ with the same parameters, and hence in $\boldsymbol{\mathcal{D}'_a}$ with parameters only the a-degrees coding the standard model (since $\mathbf{0}'_{\boldsymbol{a}}$ is obviously definable in $\boldsymbol{\mathcal{D}'_a}$).

Given $\boldsymbol{z}$, there is a set of a-degree $\boldsymbol{z}$ coded by parameters below $\boldsymbol{z} \cup \mathbf{0}'_{\boldsymbol{a}}$. If $\boldsymbol{z} \in \mathcal{X}$, then $\boldsymbol{z}$ is arithmetical in $deg_a(X)$, and thus $deg_a(X)$ is an upper bound of $\mathcal{X}$. To show that it is the least upper bound, it is enough to show that there are two a-degrees in $\mathcal{X}$ which join to $deg_a(X)$. But since below $\boldsymbol{z} \cup \mathbf{0}'_{\boldsymbol{a}}$ one can code only sets of a-degree at most $\boldsymbol{z} \cup \mathbf{0}'_{\boldsymbol{a}}$ (because the recovering procedure is arithmetical in the parameters coding a set), we know that

$$\text{if } \boldsymbol{z} \cup \mathbf{0}'_{\boldsymbol{a}} \leq deg_a(X), \text{ then } \boldsymbol{z} \in \mathcal{X}.$$

Thus it is enough to find two a-degrees $\boldsymbol{z_1}$ and $\boldsymbol{z_2}$ such that

$$\boldsymbol{z_1} \cup \boldsymbol{z_2} = deg_a(X).$$

That such a-degrees exist follows by using the method of XII.2.4 to build, as in the proof of V.2.26, two ω-generic sets A and B such that

$$A \oplus B \equiv_a deg_a(X),$$

which is possible because $deg_a(X)$ is above $\mathbf{0}'_{\boldsymbol{a}}$. $\square$

Theorem XIII.1.17 Failure of homogeneity for $\boldsymbol{\mathcal{D}}'_{\boldsymbol{a}}$ (Shore) *$\boldsymbol{\mathcal{D}}'_{\boldsymbol{a}}$ and $\boldsymbol{\mathcal{D}}'_{\boldsymbol{a}}(\geq \mathbf{0}'_{\boldsymbol{a}})$ are not elementarily equivalent.*

Proof. Consider the formula $\varphi(\boldsymbol{a})$ saying that any set coded by parameters below $\boldsymbol{a}'$ w.r.t. to a standard model of arithmetic coded below $\boldsymbol{a}'$ is arithmetical in $\emptyset^{(\omega)}$. This is a formula of Second-Order Arithmetic, and can be faithfully translated into a formula $\varphi^*(\boldsymbol{a})$ of a-degrees.

Since the *decoding* procedure (going from parameters coding a set to the set coded by them) is arithmetical, $\varphi^*(\mathbf{0}'_{\boldsymbol{a}})$ is true. Since the *coding* procedure (going from a set to parameters coding it) is also arithmetical, $\varphi^*(\mathbf{0}''_{\boldsymbol{a}})$ is false, because every set arithmetical in $\mathbf{0}''_{\boldsymbol{a}}$ can be coded by parameters below it, w.r.t. to a standard model of arithmetic coded below $\mathbf{0}''_{\boldsymbol{a}}$.

But $\mathbf{0}'_{\boldsymbol{a}}$ and $\mathbf{0}''_{\boldsymbol{a}}$ are the arithmetical jumps of the least elements of $\boldsymbol{\mathcal{D}_a}$ and $\boldsymbol{\mathcal{D}_a}(\geq \mathbf{0}'_{\boldsymbol{a}})$, respectively, and thus they are definable by the same formula in the two theories with jump. Thus the sentences $\varphi^*(\mathbf{0}'_{\boldsymbol{a}})$ and $\varphi^*(\mathbf{0}''_{\boldsymbol{a}})$ are actually the same sentence interpreted in the two theories, true in one case and false in the other. And the two theories are not elementarily equivalent. $\square$

Theorem XIII.1.18 Restriction on automorphisms of $\boldsymbol{\mathcal{D}}'_{\boldsymbol{a}}$ (Shore) *Every automorphism of $\boldsymbol{\mathcal{D}}'_{\boldsymbol{a}}$ is the identity on the cone above $\mathbf{0}'_{\boldsymbol{a}}$.*

Proof. As in the second part of V.7.22, using the analogue of V.7.10 proved in XIII.1.16. $\square$

The proofs of some of these global theorems obviously provide more than we stated. For example, we actually proved that every relation on a-degrees above $\mathbf{0}'_{\boldsymbol{a}}$ which is definable in Second-Order Arithmetic is definable in $\boldsymbol{\mathcal{D}_a}$ with $\mathbf{0}'_{\boldsymbol{a}}$ as a parameter, and that every automorphism of $\boldsymbol{\mathcal{D}_a}$ which fixes $\mathbf{0}'_{\boldsymbol{a}}$ is the identity above it.

All the global results proved so far, except for the characterization of the complexity of the theory, have been relative to the structure of a-degrees with arithmetical jump. But Slaman and Woodin [199?] have proved that *the arithmetical jump operator is definable in* $\boldsymbol{\mathcal{D}_a}$ (quite naturally, their proof is closer in spirit to the indirect definability of the ω-jump in $\boldsymbol{\mathcal{D}}$ on p. I.543, than to the direct definability of the jump in XI.5.17). This allows us to translate all the global results for $\boldsymbol{\mathcal{D}'_a}$ just proved, as well as similar ones, into results for $\boldsymbol{\mathcal{D}_a}$. In particular:

1. **definability**
   Every relation on a-degrees above $\mathbf{0}'_a$ *which is definable in Second-Order Arithmetic is definable in* $\boldsymbol{\mathcal{D}_a}$.

2. **homogeneity**
   If $\boldsymbol{\mathcal{D}_a}(\geq \boldsymbol{a})$ *is elementarily equivalent to* $\boldsymbol{\mathcal{D}_a}$*, then* $\boldsymbol{a}$ *is arithmetically low* (i.e. it has arithmetical jump $\mathbf{0}'_a$).

   If $\boldsymbol{\mathcal{D}_a}(\geq \boldsymbol{a})$ *is isomorphic to* $\boldsymbol{\mathcal{D}_a}(\geq \boldsymbol{b})$*, then* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *have the same arithmetical jump.*

3. **automorphisms**
   Every automorphism of $\boldsymbol{\mathcal{D}_a}$ *is the identity on the cone above* $\mathbf{0}'_a$.

Slaman and Woodin [199?] have also proved that *there are at most countably many automorphisms of* $\boldsymbol{\mathcal{D}_a}$, although it is not known whether there are nontrivial ones.

Finally, we provide an analogue of V.8.4. The iterations of the arithmetical jump operator are defined as for the jump operator (V.1.8 and V.1.10), and the notion of n-l.u.b. is defined in V.8.2.

Theorem XIII.1.19 (Odifreddi [1983]) $\mathbf{0}^{(\boldsymbol{\omega})}_{\boldsymbol{a}}$ *is the 1-least upper bound of* $\{\mathbf{0}^{(\boldsymbol{n})}_{\boldsymbol{a}}\}_{n\in\omega}$.

Proof. There are two things to prove:

- *if* $\boldsymbol{a}$ *is an upper bound of* $\{\mathbf{0}^{(\boldsymbol{n})}_{\boldsymbol{a}}\}_{n\in\omega}$*, then* $\mathbf{0}^{(\boldsymbol{\omega})}_{\boldsymbol{a}} \leq \boldsymbol{a}'$
  By definition of the arithmetical jump operator, this means that
  $$(\forall n)(\emptyset^{(\omega\cdot n)} \leq_a A) \Rightarrow \emptyset^{(\omega^2)} \leq_a A^{(\omega)}.$$
  If $\emptyset^{(\omega\cdot n)} \leq_a A$, then $\emptyset^{(\omega\cdot n)} \leq_T A^{(m)}$ for some m and $\emptyset^{(\omega\cdot n)} \leq_T A^{(\omega)}$. By the first part of the proof of V.8.4, then $\emptyset^{(\omega^2)} \leq_T A^{(\omega+2)}$ and thus $\emptyset^{(\omega^2)} \leq_a A^{(\omega)}$.

- *there is an upper bound* $\boldsymbol{a}$ *of* $\{\mathbf{0}_{\boldsymbol{a}}^{(\boldsymbol{n})}\}_{n\in\omega}$ *such that* $\mathbf{0}_{\boldsymbol{a}}^{(\boldsymbol{\omega})} = \boldsymbol{a}'$
  We want A such that $(\forall n)(\emptyset^{(\omega\cdot n)} \leq_a A)$ and $A^{(\omega)} \leq_a \emptyset^{(\omega^2)}$. Note that we cannot obtain $(\forall n)(\emptyset^{(\omega\cdot n)} \leq_T A)$ for such an A because otherwise, by the proof of V.8.4, $\emptyset^{(\omega^2)} \leq_T A''$ and thus $A^{(\omega)} \leq_a \emptyset^{(\omega^2)} \leq_a A$, which is a contradiction.

  We build a sequence T_n of trees arithmetically pointed and of arithmetical degree $\mathbf{0}_a^{(n)}$. By local forcing on T_n, at step n we ensure that each branch of T_{n+1} satisfies either φ_n or $\neg\varphi_n$, where $\{\varphi_n\}_{n\in\omega}$ is a recursive enumeration of the sentences $\overline{n} \in X^{(\omega)}$. Since φ_n is arithmetical, the arithmetical degree of T_n is not modified by doing this, and we obtain T_{n+1} by applying XIII.1.9.b. The construction is uniformly recursive in $\emptyset^{(\omega^2)}$, and thus if A is on $\bigcap_{n\in\omega} T_n$, we have $A^{(\omega)} \leq_T \emptyset^{(\omega^2)}$. $\Box$

Corollary XIII.1.20 $\boldsymbol{\mathcal{D}'}$ *and* $\boldsymbol{\mathcal{D}'_a}$ *are not elementarily equivalent.*

Proof. Consider the sentence φ stating that the least jump-ideal has no 1-l.u.b. This is a sentence of both $\boldsymbol{\mathcal{D}'}$ and $\boldsymbol{\mathcal{D}'_a}$, since we can quantify over ideals by quantifying over exact pairs for them (by Spector's Theorem). We know from V.8.6.b that φ does not hold in $\boldsymbol{\mathcal{D}'}$, and we have just shown that φ holds instead in $\boldsymbol{\mathcal{D}'_a}$. $\Box$

By definability of the Turing and arithmetical jumps, it follows that $\boldsymbol{\mathcal{D}}$ *and* $\boldsymbol{\mathcal{D}_a}$ *are not elementarily equivalent* (Shore [1984]). The proof of VI.5.7 shows that $\boldsymbol{\mathcal{D}_1}$*,* $\boldsymbol{\mathcal{D}_m}$*,* $\boldsymbol{\mathcal{D}_{tt}}$ *and* $\boldsymbol{\mathcal{D}_{wtt}}$ *are not elementarily equivalent to* $\boldsymbol{\mathcal{D}_a}$.

Arithmetical degrees below $\mathbf{0}'_a$

The study of $\boldsymbol{\mathcal{D}_a}(\leq \mathbf{0}'_{\boldsymbol{a}})$ is less interesting than that of $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$, for two opposite reasons.

In one sense, $\boldsymbol{\mathcal{D}_a}(\leq \mathbf{0}'_{\boldsymbol{a}})$ is easy to study because existence proofs of many results on $\boldsymbol{\mathcal{D}_a}$ automatically provide the same results below $\mathbf{0}'_a$. For example, an examination of the proof of XIII.1.6 shows that a minimal a-degree below $\mathbf{0}'_a$ was actually constructed. More generally, by an examination of the proof of the initial segment embeddings of Simpson [1985] one notices that *every lattice with a presentation arithmetical in* $\emptyset^{(\omega)}$ *is embeddable as an initial segment of* $\boldsymbol{\mathcal{D}_a}(\leq \mathbf{0}'_{\boldsymbol{a}})$. Since not every such lattice is embeddable in the T-degrees below $\emptyset^{(\omega)}$ (by a computation as in XI.3.16), one immediately obtains a nonisomorphism result, which Simpson [1985] strengthens to: *the first-order theories of* $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}^{(\boldsymbol{\omega})})$ *and* $\boldsymbol{\mathcal{D}_a}(\leq \mathbf{0}'_{\boldsymbol{a}})$ *are not elementarily equivalent.*

In another sense, $\boldsymbol{\mathcal{D}_a}(\leq \mathbf{0}'_{\boldsymbol{a}})$ is difficult to study because the proofs of many results on $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ cannot be extended, since they use approximation or domination arguments. The former rest on the Limit Lemma, which has no counterpart at the arithmetical level because limits of arithmetical functions are still arithmetical. The latter are of little use, because almost every set is dominated by an arithmetical function. To make this precise, we introduce an analogue of the notion of hyperimmune-free degree (V.5.2).

Definition XIII.1.21 *An a-degree* $\boldsymbol{a} > \mathbf{0_a}$ *is* **arithmetically hyperimmune-free** *if, for every* $A \in \boldsymbol{a}$ *and* $f \leq_a A$*,* f *is majorized by an arithmetical function.*

The next result should be contrasted with the situation for T-degrees, which (by V.5.3.c) is just the opposite.

Proposition XIII.1.22 (Sacks [1969]) *The set of arithmetically hyperimmune-free a-degrees is comeager.*

Proof. Since there are only countably many arithmetical formulas φ, and the intersection of countably many comeager sets is comeager, it is enough to show that for any given arithmetical formula φ there is an arithmetical function f such that the set

$$\mathcal{A}_\varphi = \{A \,:\, A \models (\forall x)(\exists y)\varphi(x,y) \;\Rightarrow\; A \models (\forall x)(\exists y \leq f(x))\varphi(x,y)\}$$

is comeager. Given φ, we define an arithmetical function f such that, for any given x,

$$\{A : A \models (\exists y)\varphi(\overline{x},y) \;\Rightarrow\; A \models (\exists y \leq \overline{f(x)})\varphi(\overline{x},y)\}$$

is comeager. The result follows by intersecting all these sets, for any x.

We use two basic facts, namely that for ψ arithmetical:

- $\{A : A \models \psi\}$ is either meager or comeager
- we can arithmetically tell which of the two cases happens.

Both facts are easily proved by induction on the complexity of ψ, by using the Kuratowsky-Ulam Theorem V.3.11. Thus the predicate

$$P(x,z) \Leftrightarrow \{A : A \models (\exists y \leq \overline{z})\varphi(\overline{x},y)\} \text{ is comeager}$$

is arithmetical, and hence so is the function

$$f(x) = \begin{cases} \mu z.\, P(x,z) & \text{if } (\exists z)P(x,z) \\ 0 & \text{otherwise.} \end{cases}$$

There are two cases:

1. If $(\exists z)P(x,z)$, then $P(x,f(x))$ holds and, by definition, the set
$$\{A : A \models (\exists y \leq \overline{f(x)})\varphi(\overline{x},y)\}$$
is comeager. Hence so is the set
$$\{A \,:\, A \models (\exists y)\varphi(\overline{x},y) \;\Rightarrow\; A \models (\exists y \leq \overline{f(x)})\varphi(\overline{x},y)\},$$
which is bigger because it also contains the A's not satisfying the premise.
2. If $\neg(\exists z)P(x,z)$, then the set
$$\{A : A \models (\exists y \leq \overline{z})\varphi(\overline{x},y)\}$$
is not comeager for any z, hence it is meager by the facts quoted above. Thus its complement
$$\{A : A \models (\forall y \leq \overline{z})\neg\varphi(\overline{x},y)\}$$
is comeager, for every z. By intersection
$$\{A : A \models (\forall y)\neg\varphi(\overline{x},y)\} = \{A : A \models \neg(\exists y)\varphi(\overline{x},y)\}$$
is comeager, and hence so is the bigger set
$$\{A : A \models (\exists y)\varphi(\overline{x},y) \;\Rightarrow\; A \models (\exists y \leq \overline{f(x)})\varphi(\overline{x},y)\}. \quad \square$$

When dealing with $\boldsymbol{\mathcal{D}_a(\leq 0'_a)}$, one would like to characterize the behavior of the arithmetical jump operator restricted to it. On top of the methodological difficulties just discussed, there is an additional difficulty. To even *formulate* an analogue of the Shoenfield Jump Inversion Theorem XI.1.20, one needs an analogue of the notion of (relative) recursive enumerability. This will be done in the next section, but first we prove that the answer to the jump problem is not trivial.

Proposition XIII.1.23 (Odifreddi [1983], Jockusch and Shore [1984]) *Not every a-degree $\boldsymbol{b}$ such that $\boldsymbol{0'_a} \leq \boldsymbol{b} \leq \boldsymbol{0''_a}$ is the arithmetical jump of some a-degree $\boldsymbol{a} \leq \boldsymbol{0'_a}$.*

Proof. If $A \leq_a \emptyset^{(\omega)}$, then $A^{(\omega)}$ is an arithmetical singleton. By forcing, there is a set B such that $\emptyset^{(\omega)} \leq_a B \leq_a \emptyset^{(\omega+\omega)}$, and B is not an arithmetical singleton. Since an a-degree contains an arithmetical singleton if and only if it contains only arithmetical singletons, the a-degree of B is not the arithmetical jump of any a-degree below $\boldsymbol{0'_a}$. $\square$

The complete characterization of the class of a-degrees which are arithmetical jumps of some a-degree below $\boldsymbol{0'_a}$ is given in XIII.4.6.

XIII.2 An Analogue of R.E. Sets

Recall from XII.2.21 that the ω-hop J defined by a recursive function f is the operator

$$J(X, f) = \oplus_{n \in \omega} H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X))\cdots)).$$

Moreover, ω-hops generalize the ω-jump operator in the same way as hops (XI.1.14) generalize the jump operator. As the r.e. sets are the result of hops applied to $X = \emptyset$, we have an analogue of r.e.-ness at the ω-level in the sets that can be obtained by applying ω-hops to $X = \emptyset$.

Definition XIII.2.1 (Jockusch and Shore [1984]) *The* **$\boldsymbol{\omega}$-r.e.a. sets** *are those of the form $J(\emptyset, f)$, for some recursive function f.*

More generally, the **sets $\boldsymbol{\omega}$-r.e.a. in $\boldsymbol{X}$** *are those of the form $J(X, f)$, for some recursive function f.*

The reason for the name 'ω-r.e.a.' is to stress the fact that we are actually considering (iterated) r.e. operations that move up in degree (perhaps trivially), and produce sets r.e. in and *above* given sets. The name 'ω-r.e.' is reserved for the sets at the ω-th level of the Boolean Hierarchy (IV.1.18), i.e. for the sets with a recursive approximation whose number of changes is bounded by a recursive function (equivalently, for the sets *tt*-reducible to $\mathcal{K}$). The connection between the two notions is dealt with in XI.5.10 and (part 2 of the proof of) XII.3.12.

In the remaining part of the chapter we will study the following structure.

Definition XIII.2.2 $\boldsymbol{\mathcal{R}_a}$ *is the substructure of* $\boldsymbol{\mathcal{D}_a}$ *consisting of the ω-r.e.a. a-degrees.*

Basic properties

Having introduced an analogue of r.e. sets, we now prove the analogues of some basic results: the Enumeration Theorem (II.1.5), the Fixed-Point Theorem (II.2.10), and the Completeness Theorem for $\mathcal{K}$ (III.1.2). They all actually hold not only for the ω-r.e.a. sets, but also for their relativizations (the ω-hops). The results are all quite trivial, being simple consequences of the definitions and of their original versions for r.e. sets.

The definition XII.2.21 of ω-hop uses total recursive functions. Since these cannot be recursively enumerated, there seems to be a difficulty in producing an effective enumeration of the ω-hops. But by considering undefined values as representing the trivial hop, we can easily overcome the problem.

Definition XIII.2.3 Enumeration of the $\boldsymbol{\omega}$-hops. *Let $\boldsymbol{J_e(X)}$ be the ω-hop defined by the recursive function g_e such that*

$$\mathcal{W}_{g_e(x)} = \begin{cases} \mathcal{W}_{\varphi_e(x)} & \textit{if } \varphi_e(x)\downarrow \\ \emptyset & \textit{otherwise.} \end{cases}$$

For $X = \emptyset$ we write $\boldsymbol{J_e}$ for $J_e(\emptyset)$.

It is easily verified that $\{J_e(X)\}_{e\in\omega}$ provides an enumeration of the sets ω-r.e.a. in X, uniformly in X. In particular, $\{J_e\}_{e\in\omega}$ is an enumeration of the ω-r.e.a. sets.

The next result is crucial for the development ahead, and plays the same role as its analogue did for the r.e. sets.

Proposition XIII.2.4 Fixed-Point Theorem for $\boldsymbol{\omega}$-hops. *If f is recursive, there is e such that for every X*

$$J_e(X) = J_{f(e)}(X).$$

Proof. By the Fixed-Point Theorem II.2.10. □

The next result justifies our approach. Recall that we could not introduce an analogue of r.e. sets in the most natural way, by considering the sets enumerated by arithmetical functions, since they coincide with the arithmetical sets. We thus considered the complete r.e. set $\mathcal{K}$, of which $\emptyset^{(\omega)}$ provides a natural analogue, and looked for a notion for which $\emptyset^{(\omega)}$ was a complete set.

Proposition XIII.2.5 *$X^{(\omega)}$ is 1-complete for sets ω-r.e.a. in X.*

Proof. $X^{(\omega)}$ is obviously ω-r.e.a. in X. Given a set

$$A = \oplus_{n\in\omega} H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X)\cdots))$$

ω-r.e.a. in X, then $A \leq_1 X^{(\omega)}$ because each $H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(X)\cdots))$ is uniformly 1- reducible to $X^{(n+1)}$. □

Recall that any set 1-reducible to $\mathcal{K}$ is r.e. The analogue of this fact fails here: *it is not true that any set 1-reducible to $\emptyset^{(\omega)}$ is ω-r.e.a.* We will see both arithmetical and nonarithmetical counterexamples in the following (XIII.2.9).

The analogy

$$\frac{\text{r.e.}}{\text{jump}} = \frac{\omega\text{-r.e.a.}}{\omega\text{-jump}}$$

will be shown to be perfect in Section 4, where analogues of all the theorems of Section XI.1 on the jump operator are proved. But this is also the limit of the notion of ω-r.e.a. set, since the way it has been introduced does not suggest much of a relation with the notion of arithmetical computability. We will be able to prove some important results in Sections 3 and 5, but a thorough analogy with r.e. sets and recursive computations simply fails, since *there are arithmetical sets which are not ω-r.e.a.*, as we will see in XIII.2.8.

Representation of infinite hops

Since we are going to construct ω-hops, we would like to set up a framework for this. Consider the ω-hop A defined by

$$A(X) = \oplus_{n\in\omega} H_{f(n)}(H_{f(n-1)}(\cdots (H_{f(0)}(X))\cdots)).$$

We will picture A as consisting of infinitely many columns:

- the first column will represent the input X
- the second column V_0 will be a set r.e. in X (i.e. $\mathcal{W}^X_{f(0)}$)
- the third column V_1 will be a set r.e. in the join of the previous columns (i.e. $\mathcal{W}^{X\oplus\mathcal{W}^X_{f(0)}}_{f(1)}$), and so on.

In general we start from X, and every column is r.e. in the join of the previous ones. The ω-hop is then the infinite join of all columns:

$$X \oplus V_0 \oplus V_1 \oplus \cdots$$

Given A, we will as usual indicate by $A^{[n]}$ its n-th column and by $A^{[\leq n]}$ the join of its first $n+1$ columns.

The next fact tells us that ω-hops need not be generated by r.e. steps: any fixed arithmetical bound on the step-complexity still produces ω-hops, up to a-degree.

Proposition XIII.2.6 (Simpson [1985]) *If A is a recursive union of columns, each one uniformly Σ^0_{m+1} over the previous ones (for a fixed m), then A has the same a-degree as an ω-hop B.*

Proof. Let

$$A(X) = X \oplus V_0 \oplus V_1 \oplus \cdots$$

with V_n uniformly Σ^0_{m+1} over $X \oplus (\oplus_{i<n} V_i)$. Define B as

$$X \oplus X' \oplus \cdots \oplus X^{(m)} \oplus V_0 \oplus \hat{V_0}' \oplus \hat{V_0}'' \oplus \cdots \oplus \hat{V_0}^{(m)} \oplus V_1 \oplus \hat{V_1}' \oplus \cdots$$

where $\hat{Y}$ indicates the join of the columns up to Y. Then B is an ω-hop, since any column is uniformly r.e. in the previous ones (either because it is a jump, or because 'Σ^0_{m+1} over a set' means 'r.e. in its m-jump').

$A(X) \leq_T B(X)$ obviously holds. And $B(X) \leq_a A(X)$ holds because each column of $B(X)$ is uniformly recursive in $A(X)^{(m)}$. $\square$

The complexity of ω-r.e.a. sets

The next observation bounds the complexity of ω-r.e.a. sets.

Proposition XIII.2.7 (Jockusch and Shore [1984]) *Every ω-r.e.a. set is a Π_2^0-singleton.*

Proof. Let $A = \oplus_{n\in\omega} H_{f(n)}(H_{f(n-1)}(\cdots(H_{f(0)}(\emptyset))\cdots))$. Then

$$X = A \Leftrightarrow X^{[0]} = \emptyset \wedge (\forall n)(X^{[n+1]} = \mathcal{W}_{f(n)}^{X^{[\leq n]}}),$$

and the right-hand side is a Π_2^0 expression (in X), because

$$X^{[n+1]} = \mathcal{W}_{f(n)}^{X^{[\leq n]}} \;\Leftrightarrow\; (\forall z)(z \in X^{[n+1]} \leftrightarrow z \in \mathcal{W}_{f(n)}^{X^{[\leq n]}}). \quad \Box$$

In particular, by XII.2.16 *every ω-r.e.a. set is Δ_1^1*. Obviously, $\emptyset^{(\omega)}$ is ω-r.e.a. by definition, and we will see in Volume III that every $\emptyset^{(\alpha)}$ for $\alpha < \omega_1^{ck}$ is ω-r.e.a. It follows that *the Turing degrees of ω-r.e.a. sets are cofinal in the Turing degrees of Δ_1^1 sets.*

The inclusion of the class of ω-r.e.a. sets in the class of Π_2^0-singletons is proper, as we show in XIII.2.10. Nevertheless, the proposition just proved is quite useful, and provides a natural way of defining Π_2^0-singletons. So much so that the results of the next section were originally proved as results about Π_2^0-singletons, before the notion of an ω-r.e.a. set was even isolated. It was then later realized that their proofs actually produced such sets. The reader should thus keep in mind that *all existence results of ω-r.e.a. sets are also existence results of Π_2^0-singletons.*

By using XIII.2.7 and known results on arithmetical singletons from Section XII.2 we can provide a number of counterexamples.

Corollary XIII.2.8 *Not every arithmetical set is ω-r.e.a.*

Proof. By XII.2.17.1, not every arithmetical set is a Π_2^0-singleton. $\Box$

As already noted, the existence of arithmetical sets which are not ω-r.e.a. stresses the anomalous relationship between ω-r.e.a. sets and arithmetical computations, and shows that the ω-r.e.a. sets are not full analogues of the r.e. sets.

Exercises XIII.2.9 (Jockusch and Shore [1984], Simpson [1985])

a) *Not every arithmetical T-degree is ω-r.e.a.* (Hint: from XII.2.18.1, since by XII.1.11 there are arithmetical 3-generic sets, and no such set is Δ_2^0.)

b) *ω-generic a-degrees do not bound ω-r.e.a. nonarithmetical a-degrees.* (Hint: by XII.2.18.2.) This is an analogue of the fact that 1-generic T-degrees do not bound r.e. nonrecursive T-degrees, see XI.2.3.3.

c) *Not every a-degree below $\mathbf{0}'_a$ is ω-r.e.a.* (Hint: by part b), since by XII.1.11 there are ω-generic sets recursive in $\emptyset^{(\omega)}$.)

d) *There are both arithmetical and nonarithmetical sets 1-reducible to $\emptyset^{(\omega)}$ and not ω-r.e.a.* (Hint: from part a), since every arithmetical set is 1-reducible to $\emptyset^{(\omega)}$, and by part b), since by XIII.1.11.b there are ω-generic sets 1-reducible to $\emptyset^{(\omega)}$.)

The next result shows that the arithmetical Π^0_2-singletons are a wider class than the ω-r.e.a. sets, even up to T-degree.

Proposition XIII.2.10 (Jockusch and Shore [1984]) *There is an arithmetical Π^0_2-singleton whose T-degree is not ω-r.e.a.*

Proof. First note that every Δ^0_2 set A is a Π^0_2-singleton, since

$$X = A \Leftrightarrow (\forall z)(z \in X \leftrightarrow z \in A),$$

and the right-hand side is Π^0_2. To prove the result is then enough to find $A \leq_T \mathcal{K}$ whose T-degree is not ω-r.e.a. Let A be a 1-generic set recursive in $\mathcal{K}$. Then A does not bound r.e. nonrecursive sets (XI.2.3), and thus no nonrecursive ω-r.e.a. set can be recursive in it (since any such set must have a first column that is r.e. and nonrecursive). $\Box$

Harrington [1976a] has proved that the Π^0_2-singletons recursive in $\emptyset^{(\omega)}$ are a wider class than the ω-r.e.a. sets, even up to a-degree, i.e. *there is a Π^0_2-singleton recursive in $\emptyset^{(\omega)}$ whose a-degree is not ω-r.e.a.*. The proof does not rely on ω-generic sets, since they are not implicitly definable in arithmetic.

XIII.3 An Analogue of Post's Problem

We can now start our detailed investigation of ω-r.e.a. sets. Since ω-generic sets do not have ω-r.e.a. a-degree (XIII.2.9.b), forcing techniques are not useful in this context.

The goal of this section is to prove the following result.

Theorem XIII.3.1 Solution to the analogue of Post's Problem (Harrington [1976]) *There is an ω-r.e.a. set A which is neither arithmetical nor arithmetically complete (i.e. $\emptyset^\omega \not\leq_a A$).*

The proof is quite complicated, and we will approximate it in various stages. We will also remain as informal as possible, leaving to the reader the task of formalizing the details.

The obvious strategy is to make A arithmetically low (i.e. with ω-jump $\emptyset^{\omega}$) to ensure incompleteness, and to enforce nonarithmeticity by direct action. The conditions to achieve these two goals are:

$$A(X)' \equiv_T A(X) \oplus X' \equiv_T A(X'), \tag{XIII.1}$$

uniformly in X, and

$$X <_T A(X) \tag{XIII.2}$$

where $A(X)$ is an ω-hop. Then $A = A(\emptyset)$ will be an ω-r.e.a. set such that:

- *A is arithmetically low*
  From XIII.1 we have by induction, uniformly in X and n:

  $$A(X)^{(n)} \equiv_T A(X) \oplus X^{(n)} \equiv_T A(X^{(n)}).$$

  Then $A^{(n)} \equiv_T A(\emptyset^{(n)})$ and so $A^{(n)} \leq_T \emptyset^{(\omega)}$ uniformly in n, because $A(X)$ is an ω-hop, and the m-th column of $A(\emptyset^{(n)})$ is uniformly recursive in $\emptyset^{(n+m)}$ and hence in $\emptyset^{(\omega)}$. Then $A^{(\omega)} \leq_T \emptyset^{(\omega)}$.

- *A is not arithmetical*
  Otherwise $A \leq_T \emptyset^{(n)}$ for some n, and from XIII.1

  $$A(\emptyset^{(n)}) \leq_T A \oplus \emptyset^{(n)} \leq_T \emptyset^{(n)},$$

  contradicting XIII.2.

We now turn to the construction of A. Condition XIII.1 is global on the ω-hop A and presents a first problem, because in

$$A(X)' \equiv_T A(X) \oplus X' \equiv_T A(X')$$

A plays two different roles, being applied to both X and X'. The simplest way out is to suppose that we have one of the roles and to determine the other. We may imagine that we have D and build A such that

$$A(X)' \equiv_T A(X) \oplus X' \equiv_T D(X').$$

The construction of A will be uniform in D and thus it will determine $A = J_{f(e)}$ depending on $D = J_e$, for some recursive function f. Then by the *Fixed-Point Theorem XIII.2.4* there is e such that $J_e = J_{f(e)}$. By letting $D = J_e$ for this particular e, we actually have

$$J_e(X)' \equiv_T J_e(X) \oplus X' \equiv_T J_e(X'),$$

as required. Thus the Fixed-Point Theorem allows one to control the ω-jump by a uniform control of the jump.

A first approximation to the construction

The construction of A is thus reduced to the following: given $D(X')$, build $A(X)$ such that

$$A(X)' \equiv_T A(X) \oplus X' \equiv_T D(X').$$

That is, in terms of our representation of ω-hops, given

$$D(X') = X' \oplus W_0 \oplus W_1 \oplus \cdots$$

build

$$A(X) = X \oplus V_0 \oplus V_1 \oplus \cdots$$

where, in both cases, each column is r.e. in the join of the previous ones.

A new problem is that we have to build columns of A, but the condition

$$A(X)' \equiv_T D(X')$$

is global on A. Our first goal is to build A such that

$$A^{[\leq m]}(X)' \equiv_T D^{[\leq m]}(X') \qquad \text{(XIII.3)}$$

uniformly in m and X, where $A^{[\leq m]}(X) = X \oplus (\oplus_{i \leq m} V_i)$, and similarly for $D^{[\leq m]}$. This way we control the jump of bigger and bigger initial segments of our set, while building columns. By taking joins (recall that D is the join of its columns), we obtain

$$\oplus_{m \in \omega}(A^{[\leq m]}(X)') \equiv_T D(X').$$

This is not exactly what we want. But if we also had the condition

$$A(X)' \equiv_T \oplus_{m \in \omega}(A^{[\leq m]}(X)'), \qquad \text{(XIII.4)}$$

then XIII.3 and XIII.4 together would imply $A(X)' \equiv_T D(X')$.

Let us start by attacking XIII.3. We are building A by columns, so we want V_0 to start with. We are given W_0 r.e. in X' and want V_0 r.e. in X such that

$$(X \oplus V_0)' \equiv_T X' \oplus W_0.$$

Since $X' \oplus W_0$ is a typical set r.e. in and above X', and $X \oplus V_0$ is a typical set r.e. in and above X, V_0 can be obtained by the Sacks Jump Inversion Theorem XI.1.23 (the relativization to X of the following statement: every degree r.e. in and above $\mathbf{0'}$ is the jump of an r.e. degree).

To obtain V_1 we now have W_1 r.e. in $X' \oplus W_0$, and want V_1 r.e. in $X \oplus V_0$ such that

$$(X \oplus V_0 \oplus V_1)' \equiv_T X' \oplus W_0 \oplus W_1.$$

By the first step, $X' \oplus W_0 \equiv_T (X \oplus V_0)'$, and we are then again in a position to apply Sacks Jump Inversion Theorem (relativized now to $X \oplus V_0$) and obtain V_1.

This procedure can be uniformly iterated and gives A satisfying XIII.3:

$$\begin{array}{lll} X & \mapsto & X' \\ X \oplus V_0 & \mapsto & X' \oplus W_0 \equiv_T (X \oplus V_0)' \\ X \oplus V_0 \oplus V_1 & \mapsto & X' \oplus W_0 \oplus W_1 \equiv_T (X \oplus V_0 \oplus V_1)' \\ \dots & & \end{array}$$

To satisfy XIII.2 is trivial. We want $X <_T A(X)$. Since $X \leq_T A(X)$ always holds, $A(X)$ being a union of columns one of which is X itself, this is equivalent to asking $A(X) \not\leq_T X$. For this it is enough to insert an additional column V_{-1} between X and V_0, giving a set r.e. in and strictly above X. Then do the above construction: given, for example, W_0 r.e. in X', we want V_0 such that

$$V_0 \text{ is r.e. in } X \oplus V_{-1}$$
$$(X \oplus V_{-1} \oplus V_0)' \equiv_T X' \oplus W_0.$$

Sacks Jump Inversion Theorem (relativized to $X \oplus V_{-1}$) ensures the existence of V_0 whenever $X' \equiv_T (X \oplus V_{-1})'$ (i.e. V_{-1} is low over X), since then W_0 is r.e. in $(X \oplus V_{-1})'$ and

$$(X \oplus V_{-1} \oplus V_0)' \equiv_T (X \oplus V_{-1})' \oplus W_0 \equiv_t X' \oplus W_0.$$

Thus XIII.2 is taken care of simply by *choosing V_{-1} as a set r.e. in and above X, not recursive in it, and low over X* (a relativized construction of an r.e. low nonrecursive set).

We still have to ensure XIII.4, i.e.

$$A(X)' \equiv_T \oplus_{m\in\omega}(A^{[\leq m]}(X)').$$

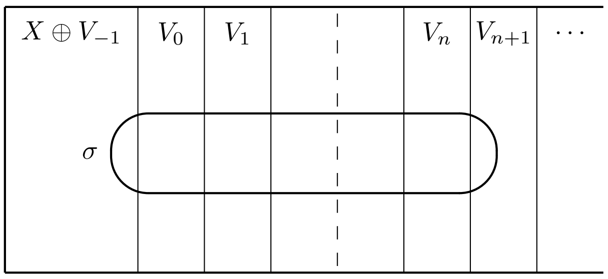
But one direction, namely $\oplus_{m\in\omega}(A^{[\leq m]}(X)') \leq_T A(X)'$, is always true, since $A^{[\leq m]}(X) \leq_T A(X)$ uniformly in m. We are thus left with the condition

$$A(X)' \leq_T \oplus_{m\in\omega}(A^{[\leq m]}(X)'),$$

which can be attacked directly by satisfying the following additional negative requirements in the construction of the columns:

$$M_e : \text{decide } \{e\}^{A(X)}(e)\downarrow\text{, recursively in finitely many columns.}$$

Note that if $\{e\}^{A(X)}(e)$ converges, then it does so by using only a finite string σ:



The idea is of course to freeze σ, but this might require action on (finitely) many columns, while our construction builds one column at a time. We will have to concoct a strategy that will be uniformly applied in the construction of every column, in such a way that the cooperative effect will be to freeze σ.

First, we can easily ensure that the additional requirements will not put too much restraint on a single column, by letting M_e act on a column V_n only for $n \geq e$. Thus each V_n will be built as in the Sacks Jump Inversion Theorem, with a finite modification also needed to satisfy the negative requirements M_e for each $e \leq n$, with highest priority.

We now analyze the action to be taken by V_n to satisfy M_e (if $e \leq n$). We already know the columns $X, V_{-1}, \ldots, V_{n-1}$ (since V_n is being built r.e. in them), but we can only approximate the next columns, by simulating their construction for a while. However, to do this we need to know the construction, that we have not yet defined. This will require *another use of the Fixed-Point Theorem.*

We refer to the picture below. Suppose σ affects the columns up to V_{n+m}: to fix σ we will have to freeze certain parts of these columns (say those up to $\tau(m)$, if $\{e\}^{\sigma}(e)$ converges in at most $\tau(m)$ steps). To do this we need to simulate the construction of V_{n+m} up to a stage $\tau(m)$, and be sure that the real construction of V_{n+m} will freeze it. Now $V_{n+m,\tau(m)}$ is defined using the oracle on the previous columns only up to $\tau(m)$ (by convention), but V_{n+m-1} has not yet necessarily settled up to $\tau(m)$ at this stage, and it might do so only at a stage $\tau(m-1) > \tau(m)$. The construction of V_{n+m-1} will try to freeze this column up to $\tau(m-1)$, so that $V_{n+m,\tau(m)}$ will be final up to $\tau(m)$. By going backwards we see that we need to freeze bigger and bigger segments, until we obtain a segment $\tau(0)$ for V_n. Thus we need a finite string τ such that

$$\tau(0) > \tau(1) > \cdots > \tau(m)$$
$$\{e\}^{A_\tau^{[\leq n]}(X)}(e) \text{ converges in at most } \tau(m) \text{ steps,}$$

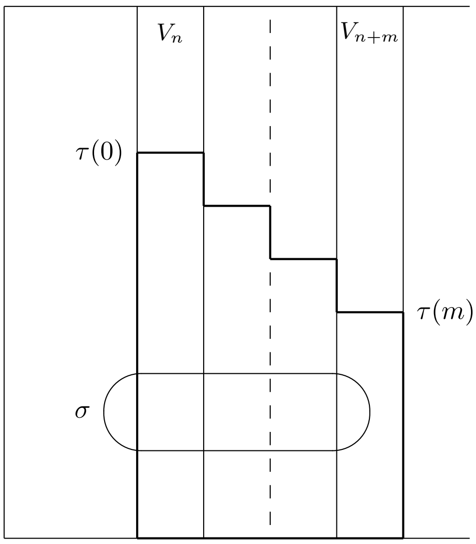
where $A_\tau^{[\leq n]}(X)$ is the approximation of $A(X)$ obtained by:

- taking the first columns to be $X \oplus V_{-1}, V_0, \ldots, V_{n-1}$
- running the construction of V_n for $\tau(0)$ stages over the previous columns
- running the construction of V_{n+1} for $\tau(1)$ stages over the first columns and the approximation to V_n just obtained, and so on.

The requirement M_e can then be rephrased, relatively to the construction of V_n, as $M_e^{V_n}$:

> at stage $s+1$ restrain y from entering V_n if y is not yet in, and there is a τ as above with $s = \tau(0)$ and $y < \tau(0)$, with $\tau(0)$ minimal.

The reason why $M_e^{V_n}$ acts at stage $\tau(0)+1$ is that we want to freeze a certain situation, and we can only refer to a current state of affairs (if we acted to freeze at a later stage, the situation might have changed by then).



Note that if M_e acts in the construction of V_n for a certain τ, then it also acts in the construction of V_{n+1} for $\langle \tau(1), \ldots, \tau(m) \rangle$, if $|\tau| > 1$. This is because V_n and $V_{n,\tau(0)}$ agree up to $\tau(0)$ (since V_n freezes the column at stage $\tau(0)$, and it prevents elements less than $\tau(0)$ from entering the column at future stages). Note also that M_e will act in the construction of V_{n+1} at stage $\tau(1) < \tau(0)$, i.e. earlier than it did for V_n, but in a different construction (that takes place afterwards).

This produces $A(X)' \leq_T \oplus_{m\in\omega}(A^{[\leq m]}(X)')$. Indeed, to decide whether $\{e\}^{A(X)}(e)$ converges we look at the construction of V_e (the first column in which M_e is considered), and note that $\{e\}^{A(X)}(e)\downarrow$ if and only if

- $\{e\}^{A(X)}(e)$ converges in at most $e+1$ steps, or
- $(\exists\tau)(\tau(0) > \cdots > \tau(m) \wedge \{e\}^{A_\tau^{[\leq e]}(X)}(e)\downarrow$ in at most $\tau(m)$ steps), where $m = |\tau|$.

Thus $\{e\}^{A(X)}(e)\downarrow$ can be decided by an existential question over the columns

$$X \oplus V_{-1} \oplus V_0 \oplus \cdots \oplus V_{e-1} = A^{[\leq e-1]}(X),$$

and it is then recursive in $A^{[\leq e-1]}(X)'$, uniformly in e. This ensures XIII.4.

The real construction

The construction just sketched provides us with a set A such that

$$A(X)' \equiv_T A(X') \text{ and } X <_T A(X).$$

While the second property is condition XIII.2, the first property is just an incomplete version of condition XIII.1, missing the part relative to $A(X) \oplus X'$. Unfortunately we used this part to show that condition XIII.2 implies that A is not arithmetical.

Exercise XIII.3.2 *There is an arithmetical (even an r.e.) operator A satisfying the two properties above.* Thus the original stronger formulation of condition XIII.2 is necessary. (Hint: let f be a recursive function, obtained by the uniformities of the proof of Sacks Jump Inversion Theorem, such that $X \oplus \mathcal{W}_{f(e)}^X$ is not recursive in X and

$$(X \oplus \mathcal{W}_{f(e)}^X)' \equiv_T X' \oplus \mathcal{W}_e^{X'}.$$

Choose e as a fixed-point of f and let $A(X) = X \oplus \mathcal{W}_e^X$.)

We then have to revisit the construction of A and try to satisfy the full condition XIII.1. The first idea would be to attempt a control column by column, as before. The condition is now

$$A^{[\leq m]}(X)' \equiv_T A^{[\leq m]}(X) \oplus X' \equiv_T D^{[\leq m]}(X'),$$

which will imply, as above, that

$$\oplus_{m\in\omega}(A^{[\leq m]}(X)') \equiv_T A(X) \oplus X' \equiv_T D(X').$$

Let us start once again with the construction of V_0. Given W_0 r.e. in X', we would like to have V_0 r.e. in X such that

$$(X \oplus V_0)' \equiv_T X \oplus V_0 \oplus X' \equiv_T X' \oplus W_0.$$

But now the middle term is simply T-equivalent to X', since both X and V_0 are recursive in X' (recall that V_0 is r.e. in X). Then the last equivalence says

$$X' \equiv_T X' \oplus W_0.$$

But this is a condition on W_0, which is a given set, and does not necessarily satisfy it (when it is strictly above X', it actually does not).

There is thus no hope of using just one column to satisfy both equivalences. In general we will need two to control things separately. Then A will be built two columns at a time, and the new conditions become

$$A^{[\leq 2m]}(X)' \equiv_T A^{[\leq 2m]}(X) \oplus X' \equiv_T D^{[\leq m]}(X') \qquad \text{(XIII.5)}$$

uniformly in m and X, and

$$A(X)' \leq_T \oplus_{m\in\omega}(A^{[\leq 2m]}(X)'). \qquad \text{(XIII.6)}$$

They imply, as before, that

$$A(X)' \equiv_T A(X) \oplus X' \equiv_T D(X').$$

Once again we turn to the first step of the construction. We are still given W_0 r.e. in X', but now we want two sets V_0 and V_1 such that

V_0 is r.e. in X
V_1 is r.e. in $X \oplus V_0$
$(X \oplus V_0 \oplus V_1)' \equiv_T X' \oplus V_0 \oplus V_1 \equiv_T X' \oplus W_0$.

This will require an extension of Sacks Jump Inversion Theorem, which we formulate as follows.

Proposition XIII.3.3 *ZBC* Lemma. *Given W r.e. in Z', there are B and C such that:*

1. *B is r.e. in Z*
2. *C is r.e. in $Z \oplus B$*
3. *$(Z \oplus B \oplus C)' \equiv_T Z' \oplus B \oplus C \equiv_T Z' \oplus W$.*

As before, we can show that an iteration of the Lemma produces A satisfying XIII.5:

$$\begin{array}{lll}
X & \mapsto & X' \\
X \oplus V_0 \oplus V_1 & \mapsto & X' \oplus W_0 \equiv_T (X \oplus V_0 \oplus V_1)' \\
X \oplus V_0 \oplus V_1 \oplus V_2 \oplus V_3 & \mapsto & X' \oplus W_0 \oplus W_1 \equiv_T (X \oplus V_0 \oplus V_1 \oplus V_2 \oplus V_3)' \\
\ldots & &
\end{array}$$

Our plan is now to prove the ZBC Lemma. A modification of it in the style given above for the original Sacks Jump Inversion Theorem will take care of XIII.6, and will thus give us A as required (note that condition XIII.2 can be taken care of by inserting the additional column V_{-1} as before).

The basic module of the construction

We prove the ZBC Lemma in unrelativized form, since the complications introduced by relativization are purely notational. What we want is thus:

Proposition XIII.3.4 Unrelativized ZBC Lemma. *Given W r.e. in and above $\mathcal{K}$, there are B and C such that*

1. *B is r.e.*
2. *C is r.e. in B*
3. *$(B \oplus C)' \equiv_T \mathcal{K} \oplus C \equiv_T W$.*

Note that we would actually need $\mathcal{K} \oplus B \oplus C$ in the last equation, but this is Turing equivalent to $\mathcal{K} \oplus C$, since B is r.e.

Given W we use the fact that it is r.e. in $\mathcal{K}$ (and hence Σ^0_2) to obtain S r.e. such that

$$\begin{array}{lll}
e \in W & \Rightarrow & S^{[e]} \text{ finite} \\
e \notin W & \Rightarrow & S^{[e]} = \omega.
\end{array}$$

Recall that to obtain B r.e. such that $B' \equiv_T W$ by the Sacks Jump Inversion Theorem we would have to ensure:

- *positive action*

  $$P_e \quad : \quad B^{[e]} =^* S^{[e]}.$$

  Then B is a thick subset of S, and

  $$e \in W \;\Leftrightarrow\; \lim B^{[e]} = 0,$$

  so that $W \leq_T B'$ by the Limit Lemma.

- *negative action*

$$N_e \quad : \quad (\exists_\infty s)(\{e\}_s^{B_s}(e)\downarrow) \;\Rightarrow\; \{e\}^B(e)\downarrow .$$

  This tends to freeze converging computations from B (i.e. membership in B'). We cannot succeed in general, otherwise B would be low (contradicting the positive action), but the imposed restraints will make B' as low as possible, i.e. $B' \leq_T W$.

However, we do not want $B' \equiv_T W$, but rather

$$W \leq_T \mathcal{K} \oplus C \leq_T (B \oplus C)' \leq_T W.$$

Note that $\mathcal{K} \oplus C \leq_T (B \oplus C)'$ automatically. Thus we only have to ensure

$$\begin{aligned} W &\leq_T \mathcal{K} \oplus C \\ (B \oplus C)' &\leq_T W. \end{aligned}$$

This will be achieved, for reasons to be clarified in the construction of C, by breaking down the first condition, and thus ensuring that

$$\begin{aligned} W &\leq_T B' \\ B' &\leq_T \mathcal{K} \oplus C \\ (B \oplus C)' &\leq_T W. \end{aligned}$$

Construction of *B*. As in the sketch above, we have the requirements

$$\begin{aligned} P_e^B &\quad : \quad B^{[e]} =^* S^{[e]} \\ N_e^B &\quad : \quad (\exists_\infty s)(\{e\}_s^{B_s \oplus C}(e)\downarrow) \;\Rightarrow\; \{e\}^{B \oplus C}(e)\downarrow . \end{aligned}$$

First, note that the construction of B depends on C, which will itself be defined using B. Thus at any given stage s we only have a set C^s, which is constructed as C, but with oracle B_s in place of B. But even this set cannot be used directly, being an r.e. set. Only recursive approximations C_t^s of it can be used.

There is, however, *no appeal to the Fixed-Point Theorem* here, since C is defined using B but independently of it. We thus know its construction and can apply it to any oracle, to obtain approximations of C.

The negative requirements N_e^B act as follows. At stage $s+1$, given B_s, we run the construction of C^s for at most s stages, and for the first $t \leq s$ such that $\{e\}_s^{B_s \oplus C_t^s}(e)\downarrow$ we try to fix this computation. To do this we have to freeze the elements that have been used negatively in the computation. Those on the B-side can be restrained directly and this makes

$$\{e\}_s^{B_s \oplus C_t^s}(e) \simeq \{e\}^{B \oplus C_t^s}(e).$$

To fix the C-side, first note that (by convention) C_t^s can look at the oracle only up to t. If we thus freeze B_s up to t (i.e. the part of the oracle to which C_t^s has access), then $C_t^s = C_t$ and

$$\{e\}_s^{B_s \oplus C_t^s}(e) \simeq \{e\}^{B \oplus C_t}(e).$$

This is the best we can do, since we can only work on B_s, and once this is given, then C^s is completely determined. So, if some element is used negatively in C_t^s, we have no direct power to keep it out of C^s.

This means that we might have to drop t later and consider a bigger stage, once we discover that C_t^s is not final below the use of the computation, i.e. if some forbidden element enters C^s. But this may happen only finitely often, since we eventually reach a stage that is bigger than the use, and a true stage in the enumeration of C^s. After such a stage, C^s can no longer change below the use. Then each N_e^B produces only finite lim inf of the restraint.

The construction of B consists of putting into it an element $\langle e, x\rangle$ at stage $s+1$ whenever the positive requirement P_e^B says so (i.e. if $x \in S^{[e]}$), unless negative requirements of higher priority restrain it.

This is a usual *infinite injury argument* in the style of the Thickness Lemma.

Construction of *C*. Since we only want C r.e. in B, we may use B in the construction as an oracle, and thus C will be obtained uniformly from B. The conditions for C are

$$B' \leq_T \mathcal{K} \oplus C$$
$$(B \oplus C)' \leq_T W.$$

To satisfy the first we code the jump of B into C. Since B' is controlled by the columns of B, we attempt to code into C the place where the columns stop switching:

$$P_e^C \ : \ \text{if } (\exists z > x)(x \in B^{[e]} \leftrightarrow z \notin B^{[e]}), \text{ then put } \langle e, x\rangle \text{ into } C.$$

To control the jump we impose

$$N_e^C \ : \ (\exists_\infty s)(\{e\}_s^{B \oplus C_s}(e)\downarrow) \Rightarrow \{e\}^{B \oplus C}(e)\downarrow .$$

The negative requirements produce the usual restraint. They forbid putting in C elements that are not in it and have been used in a convergent computation that we want to preserve.

The construction consists of putting an element $\langle e, x\rangle$ in C at stage $s+1$ whenever the positive requirement P_e^C says so, i.e. if we see $x < z \leq s$ such that $x \in B^{[e]} \leftrightarrow z \notin B^{[e]}$, unless negative requirements of higher priority restrain it.

This is a usual *finite injury argument* because, by construction, each column of B is either finite or cofinite, and so it switches only finitely many times. Then the P_e^C's are finitary.

Note that ensuring $W \leq_T \mathcal{K} \oplus C$ directly would be more complicated. Just modifying the construction by substituting S for B would not work, since it would only give C r.e. in $S \oplus B$. Thus, in a sense, B and C cooperate not only to keep $(B \oplus C)'$ down, as they have to. They also keep $\mathcal{K} \oplus C$ up, which in principle should be a concern of C alone.

The ingredients of the construction

The *ZBC* Lemma produces two sets, one by *finite injury* and the other by *infinite injury*. By an *infinite iteration*, we build (two columns at a time) an ω-hop satisfying

$$\oplus_{m\in\omega}(A^{[\leq 2m]}(X)') \equiv_T A(X) \oplus X' \equiv_T D(X').$$

However, each time the *ZBC* Lemma is applied with *local variations*, to take care of an increasing number of finitely many additional requirements needed to satisfy

$$A(X)' \equiv_T \oplus_{m\in\omega}(A^{[\leq 2m]}(X)').$$

This requires looking ahead to columns of A that have not yet been built, and is justified by an appeal to the Fixed-Point Theorem. Having

$$A(X)' \equiv_T A(X) \oplus X' \equiv_T D(X'),$$

the Fixed-Point Theorem is applied once again to make A and D coincide, and thus A is actually obtained by a *double use of the Fixed-Point Theorem.*

An analogue of the Friedberg-Muchnik Theorem

With a simple trick due to Simpson [1985], we can actually extend the solution to Post's Problem just given, to an analogue of the Friedberg-Muchnik Theorem.

Theorem XIII.3.5 (Harrington [1976]) *There are two arithmetically incomparable ω-r.e.a. a-degrees.*

Proof. The previous theorem shows how to build two ω-r.e.a. sets A and B that are arithmetically incomplete and not arithmetical. If we can make them join to $\emptyset^{(\omega)}$, then they are also automatically arithmetically incomparable.

As usual, we will reduce the requirement on the ω-jump $\emptyset^{(\omega)}$ to a uniform requirement on the jump. That is, we will code $\emptyset^{(\omega)}$ into $A \oplus B$ by coding X' into $A(X) \oplus B(X)$, uniformly in X. To achieve this, recall that we ensured that

$$A(X) \oplus X' \equiv_T A(X') \quad \text{and} \quad B(X) \oplus X' \equiv_T B(X'), \qquad \text{(XIII.7)}$$

and that we could code in column V_{-1} of A and B any set r.e. in and above X, not recursive in it, and low over X. We decide to code in the V_{-1} columns of A and B the two halves of a Sacks splitting of X', i.e. sets $\mathcal{W}^X_{e_0}$ and $\mathcal{W}^X_{e_1}$ such that, uniformly in X:

- $X <_T \mathcal{W}^X_{e_0}, \mathcal{W}^X_{e_1}$
- $\mathcal{W}^X_{e_0}$ and $\mathcal{W}^X_{e_1}$ are low over X
- $\mathcal{W}^X_{e_0} \oplus \mathcal{W}^X_{e_1} \equiv_T X'$.

A relativized version of a Sacks splitting of $\mathbf{0'}$ into two low r.e. degrees produces this, see X.6.15.

We now check that $\emptyset^{(\omega)}$ can be computed from $A \oplus B$. First, for any X

$$X' \leq_T A(X) \oplus B(X), \qquad \text{(XIII.8)}$$

since X' is the join of the two V_{-1} columns of $A(X)$ and $B(X)$. Then

$$\begin{array}{lll} X'' & \leq_T & A(X') \oplus B(X') & \text{by XIII.8 applied to } X' \\ & \leq_T & X' \oplus A(X) \oplus B(X) & \text{by XIII.7} \\ & \leq_T & A(X) \oplus B(X) & \text{by XIII.8.} \end{array}$$

Iteration gives, uniformly for every n,

$$X^{(n)} \leq_T A(X) \oplus B(X)$$

and thus

$$X^{(\omega)} \leq_T A(X) \oplus B(X).$$

For $X = \emptyset$, we obtain $\emptyset^{(\omega)} \leq_T A \oplus B$, as required. $\square$

Exercise XIII.3.6 *Build a-incomparable ω-r.e.a. sets by a direct diagonalization.* (Harrington [1976]) (Hint: this uses a technique introduced later, for the next two sections. Basically, build two sequences of ω-hops $(A_n, B_n)_{n\in\omega}$ such that

$$A_n(X)' \equiv_T A_n(X) \oplus X' \equiv_T A_{n+1}(X')$$
$$B_n(X)' \equiv_T B_n(X) \oplus X' \equiv_T B_{n+1}(X'),$$

and satisfy one requirement for diagonalization at each level.)

The proof given above shows that $\mathbf{0}'_{\mathbf{a}}$ *can be split into two incomparable ω-r.e.a. a-degrees.* We do not know whether the analogue of the Splitting Theorem holds in general for ω-r.e.a. a-degrees. Related to this, we do not know whether there are minimal ω-r.e.a. a-degrees.

XIII.4 An Analogue of the Jump Classes

We have seen in Chapter XI that the results on the jump operator, i.e. the existence of $\text{Low}_{n+1} - \text{Low}_n$, $\text{High}_{n+1} - \text{High}_n$ and intermediate degrees (XI.1.19), as well as Sacks Jump Inversion Theorem (XI.1.23) could be obtained from (the relativizations of) the following two results:

1. the existence of a nonrecursive low r.e. set
2. the fact that for every hop H_i there is a nonrecursive r.e. set A such that $H_i(A) \equiv_T \emptyset'$, with A obtained uniformly from H_i (XI.1.17).

We already have the analogue of 1 for ω-r.e.a. sets (by the proof of XIII.3.1), and thus we only need to obtain the analogue of 2.

The main result

We now prove the analogue of 2, by relying on the proof of XIII.3.1 and indicating the necessary changes.

Proposition XIII.4.1 (Simpson [1985]) *For any ω-hop $E(X)$, there is an ω-r.e.a. nonarithmetical set A such that $E(A) \equiv_T \emptyset^{(\omega)}$, with A obtained uniformly from E.*

Proof. Note that the existence of an ω-r.e.a. nonarithmetical and arithmetically low set (obtained in Section 3) is a special case of this result, for $E(X) = X^{(\omega)}$. That result was obtained by ensuring that

$$A(X)' \equiv_T A(X) \oplus X' \equiv_T A(X'),$$

uniformly in X. This was enough to control the ω-jump of $A(X)$, because the ω-jump is simply an infinite iteration of the same hop (namely, the jump operator). Now, however,

$$E(A(X)) = \oplus_{n\in\omega} H_{f(n)}(\cdots(H_{f(0)}(X))\cdots), \qquad \text{(XIII.9)}$$

and in general the hops $H_{f(n)}$ are all different.

We can still hope to be able to modify the construction of XIII.3.1 by replacing the jump by any hop H_a and obtain

$$H_a(A(X)) \equiv_T A(X) \oplus X' \equiv_T A(X').$$

But this would take care of just one step of the action of E. Certainly we cannot hope to have A satisfying the condition for more than one index a at a time, since we do not have any control over H_a.

The way to override the problem is to *build a sequence* $\{A_n\}_{n\in\omega}$ *of* ω*-hops*, each one satisfying one condition at a time:

$$H_{f(n)}(A_n(X)) \equiv_T A_n(X) \oplus X' \equiv_T A_{n+1}(X').$$

Suppose the hops were degree-invariant (i.e. they gave sets with the same T-degree when applied to sets of the same T-degree). Then we would obtain the required result by taking $A = A_0$. Indeed, we have

$$\begin{array}{rcl} H_{f(0)}(A_0(X)) & \equiv_T & A_1(X') \\ H_{f(1)}(A_1(X')) & \equiv_T & A_2(X''). \end{array}$$

If $H_{f(1)}$ is degree-invariant, this implies

$$H_{f(1)}(H_{f(0)}(A_0(X))) \equiv_T A_2(X'').$$

In general, if all the $H_{f(n)}$ are degree-invariant, then

$$H_{f(n)}(\cdots(H_{f(0)}(A_0(X))\cdots) \equiv_T A_{n+1}(X^{(n+1)}). \qquad \text{(XIII.10)}$$

But, uniformly in n,

$$X^{(n+1)} \leq_T A_{n+1}(X^{(n+1)}) \leq_T X^{(\omega)},$$

the first part because $X^{(n+1)}$ is coded in $A_{n+1}(X^{(n+1)})$ as its first column, and the second part because A_{n+1} is an ω-hop, hence $A_{n+1}(X^{(n+1)})$ is recursive in $X^{(n+1+\omega)} \equiv_T X^{(\omega)}$. This would imply, by XIII.9 and XIII.10, that

$$E(A_0(X)) \equiv_T X^{(\omega)}$$

and, in particular,

$$E(A_0) \equiv_T \emptyset^{(\omega)}.$$

The problem that *the hops are not degree-invariant* can be solved quite easily. The construction of A_0 will produce

$$H_{f(0)}(A_0(X)) \equiv_T A_1(X') \qquad \text{(XIII.11)}$$

directly. To obtain

$$H_{f(1)}(H_{f(0)}(A_0(X))) \equiv_T A_2(X''),$$

we simply build A_1 such that

$$H_a(A_1(X)) \equiv_T A_2(X'),$$

for a such that

$$H_{f(1)}(H_{f(0)}(A_0(X))) \equiv_T H_a(A_1(X')).$$

Note that such an a certainly exists, by XIII.11:

$$\begin{tikzcd}
H_{f(0)}(A_0(X)) \arrow[d] & A_1(X') \arrow[l] \arrow[dl] \\
H_{f(1)}(H_{f(0)}(A_0(X))) \equiv_T H_a(A_1(X')) &
\end{tikzcd}$$

The only trouble is that to find a we need to know A_1, which is what we have to build. Thus the construction of A_1 (and, in general, of A_n) requires a *first use of the Fixed-Point Theorem*, to be able to tell in advance which index we are going to use in the construction.

We have thus reduced the whole construction to a sequence of constructions, for an array $\{A_n\}_{n\in\omega}$ of ω-hops satisfying

$$H_{a_n}(A_n(X)) \equiv_T A_{n+1}(X'),$$

for appropriate indices a_n given uniformly in n. As in Section 3, we will build A_n satisfying

$$H_{a_n}(A_n(X)) \equiv_T D_n(X'),$$

for given ω-hops $\{D_n\}_{n\in\omega}$. A *second use of the Fixed-Point Theorem* will allow us to obtain $D_n = A_{n+1}$.

Note that both times we actually apply the *Fixed-Point Theorem with parameters* (II.2.11.a), since we have to straighten up a whole sequence of constructions, instead of a single one.

We have thus reduced the construction to a class of similar constructions, of the kind: *given an ω-hop D and an index a, build a nonarithmetical ω-r.e.a. set A such that*

$$H_a(A(X)) \equiv_T A(X) \oplus X' \equiv_T D(X').$$

To achieve this we need the following modifications to the construction given in Section 3:

1. The perturbations of the *ZBC* Lemma required to satisfy
$$A(X)' \leq_T \oplus_{m\in\omega}(A^{[\leq 2m]}(X)')$$
now have the new goal
$$H_a(A(X)) \leq_T \oplus_{m\in\omega}(A^{[\leq 2m]}(X)').$$
This produces only notational modifications, since we have now to satisfy the requirements
$$M_e \quad : \quad \text{decide } \{a\}^{A(X)}(e)\!\downarrow .$$
Indeed $H_a(A(X)) = A(X) \oplus \mathcal{W}_a^{A(X)}$, and $A(X) \leq_T \oplus_{m\in\omega}(A^{[\leq 2m]}(X)')$ holds automatically. Thus we only have to control $\mathcal{W}_a^{A(X)}$.

   Recall that the perturbations required to satisfy the M_e's require a look-ahead, and hence a *third use of the Fixed-Point Theorem*.

2. While the condition
$$\oplus_{m\in\omega}(A^{[\leq 2m]}(X)') \leq_T A(X)'$$
was automatic, we now also have to ensure that
$$\oplus_{m\in\omega}(A^{[\leq 2m]}(X)') \leq_T H_a(A(X)). \qquad \text{(XIII.12)}$$
This is most easily achieved by satisfying
$$X' \leq_T H_a(A(X)), \qquad \text{(XIII.13)}$$
since (as $A(X) \leq_T H_a(A(X))$ by the definition of hop) we then have
$$A(X) \oplus X' \leq_T H_a(A(X)),$$
and the rest of the construction already ensures that
$$\oplus_{m\in\omega}(A^{[\leq 2m]}(X)') \leq_T A(X) \oplus X'. \qquad \text{(XIII.14)}$$
We thus obtain XIII.12 by transitivity from XIII.14 and XIII.13.

   Recall that the role of C in the *ZBC* Lemma was already that of coding. Thus we decide to code X' in the C-columns of A (i.e. the columns V_{2m+1}), one element at a time. To avoid interference with the rest of the coding, we shift the previous coding one column to the right. That is, we now have
$$P_e^C \quad : \quad \text{if } (\exists z > x)(x \in B^{[e]} \leftrightarrow z \notin B^{[e]}), \text{ then put } \langle e+1, x\rangle \text{ into } C.$$

If we are applying the *ZBC* Lemma to obtain V_{2m} and V_{2m+1}, then we reserve the 0-th column of C to code whether m is an element of X'. We thus have a single additional requirement:

$$P_{-1}^{C} \quad : \quad \text{if } m \in X', \text{ put the first unrestrained element } \langle 0, x\rangle \text{ into } C.$$

We cannot use X' directly, since we are building $A(X)$, but we do have X available as the first column, and thus we can use procedures recursive in X. Fix an enumeration $\{k_s\}_{s\in\omega}$ of X' recursive in X. If we discover that $m = k_s$ at stage $s+1$, then we will put into C the first $\langle 0, x\rangle$ not restrained by requirements of higher priority.

Since the priority order in the construction of V_{2m+1} is

$$M_0 > \cdots > M_m > P_{-1} > P_0 > N_0 > P_1 > N_1 > \cdots$$

(recall that the M_e's have highest priority), only the M_e's can restrain something from entering the 0-th column of C.

To show that $X' \leq_T H_a(A(X))$, fix m. Then by construction

$$m \in X' \Leftrightarrow \text{the 0-th column of } V_{2m+1} \text{ is not empty.}$$

This expression is only recursive in $A(X)'$, but it can be made recursive in $H_a(A(X))$ by noting that, by construction,

$$m \in X' \Leftrightarrow \text{the 0-th column of } V_{2m+1} \text{ is not empty below } r(m),$$

where $r(m)$ is 1 plus the total restraint imposed by $M_0, \ldots, M_m$. We can compute the total restraint by first deciding, recursively in $\mathcal{W}_a^{A(X)}$, whether M_e with $e \leq m$ is going to act (namely, using elements of V_{2m+1} negatively if $\{a\}^{A(X)}(e)\downarrow$) and, if so, by running the construction until the restraint is imposed. $\square$

Jump classes

The next results show that each jump class is not empty and it actually contains members of ω-r.e.a. degree. Since we do not know how to control the ω-jump directly, as in XI.1.22.b, we proceed indirectly as in XI.1.19.

Theorem XIII.4.2 Hierarchy Theorem (Simpson [1985])

1. *For each n, there is an ω-r.e.a. set A of $Low_{n+1} - Low_n$ a-degree, i.e. such that*

$$A^{(\omega\cdot(n+1))} \equiv_a \emptyset^{(\omega\cdot(n+1))} \quad \textit{but} \quad A^{(\omega\cdot n)} \not\equiv_a \emptyset^{(\omega\cdot n)}.$$

2. *For each n, there is an ω-r.e.a. set A of $High_{n+1} - High_n$ a-degree, i.e. such that*
$$A^{(\omega\cdot(n+1))} \equiv_a \emptyset^{(\omega\cdot(n+2))} \quad \textit{but} \quad A^{(\omega\cdot n)} \not\equiv_a \emptyset^{(\omega\cdot(n+1))}.$$

3. *There is an ω-r.e.a. set A of intermediate a-degree, i.e. such that for every n*
$$\emptyset^{(\omega\cdot n)} <_a A^{(\omega\cdot n)} <_a \emptyset^{(\omega\cdot(n+1))}.$$

Proof. As in XI.1.19, using XIII.3.1 and XIII.4.1. □

Exercise XIII.4.3 *There are incomparable ω-r.e.a. a-degrees in each jump class.* (Simpson [1985]) (Hint: use the trick of XIII.3.5 for incomparability.)

Jump inversion

We have seen in XI.1.20 and XI.1.23 that the range of the jump operator restricted to T-degrees below $\mathbf{0'}$ is the set of degrees r.e. in and above $\mathbf{0'}$, and that each of these degrees is realized as the jump of an r.e. degree.

We now provide a similar characterization of the range of the ω-jump operator restricted to a-degrees below $\mathbf{0'_a}$ (which, we recall, is the a-degree of $\emptyset^{(\omega)}$), as the set of a-degrees ω-r.e.a. in $\emptyset^{(\omega)}$.

That the condition is necessary is immediate.

Proposition XIII.4.4 (Jockusch and Shore [1984], Simpson [1985])

1. *If $A \leq_T \emptyset^{(\omega)}$, then $A^{(\omega)}$ is ω-r.e.a. in $\emptyset^{(\omega)}$ up to 1-degree.*

2. *If $A \leq_a \emptyset^{(\omega)}$, then $A^{(\omega)}$ is ω-r.e.a. in $\emptyset^{(\omega)}$ up to a-degree.*

Proof. Let $A \leq_T \emptyset^{(\omega)}$ and consider
$$\emptyset^{(\omega)} \oplus A^{(\omega)} = \emptyset^{(\omega)} \oplus A \oplus A' \oplus \cdots .$$
This is ω-r.e.a. in $\emptyset^{(\omega)}$, because A is r.e. in $\emptyset^{(\omega)}$ and $A^{(n+1)}$ is r.e. in $A^{(n)}$. Since $\emptyset^{(\omega)} \leq_1 A^{(\omega)}$ always holds, and $\emptyset^{(\omega)}$ is a cylinder,
$$A^{(\omega)} \equiv_1 \emptyset^{(\omega)} \oplus A^{(\omega)}$$
(note that from $B \leq_1 C$ we can only deduce $C \equiv_m B \oplus C$ in general, but we do have $C \equiv_1 B \oplus C$ when C is a cylinder, by VI.6.3).

Similarly, if $A \leq_a \emptyset^{(\omega)}$, then $A \leq_T \emptyset^{(\omega+n)}$ for some n. Consider
$$\emptyset^{(\omega+n)} \oplus A^{(\omega)} \equiv_1 \emptyset^{(\omega)} \oplus \emptyset^{(\omega+1)} \oplus \cdots \oplus \emptyset^{(\omega+n)} \oplus A \oplus A' \oplus \cdots .$$

This is ω-r.e.a. in $\emptyset^{(\omega)}$. But we have $\emptyset^{(\omega)} \equiv_a \emptyset^{(\omega+n)}$ and $\emptyset^{(\omega)} \leq_a A^{(\omega)}$, so

$$A^{(\omega)} \equiv_a \emptyset^{(\omega+n)} \oplus A^{(\omega)}. \quad \Box$$

Simpson [1985] shows that is not true in general that if $A \leq_a \emptyset^{(\omega)}$, then $A^{(\omega)}$ is ω-r.e.a. in $\emptyset^{(\omega)}$ up to T-degree.

Theorem XIII.4.5 (Simpson [1985]) *For any set C ω-r.e.a. in $\emptyset^{(\omega)}$ there is an ω-r.e.a. set A such that $A^{(\omega)} \equiv_T C$.*

Proof. As in XI.1.23, using XIII.4.1. $\Box$

The next result now follows by putting the pieces together.

Corollary XIII.4.6 Arithmetical Jump Inversion Theorem for $\mathcal{D}_a(\leq \mathbf{0}'_a)$ (Simpson [1985]) *The range of the arithmetical jump operator restricted to $\mathcal{D}_a(\leq \mathbf{0}'_a)$ is the set of a-degrees ω-r.e.a. in $\emptyset^{(\omega)}$.*

Notice that the results above provide more, namely similar Jump Inversion Theorems for the ω-r.e.a. a-degrees on the one hand, and for T-degrees below $\mathbf{0}^{(\omega)}$ on the other hand.

XIII.5 Comparison with the R.E. Degrees

From a methodological point of view, we have proved our results on ω-r.e.a. sets by a control of the ω-jump. This was useful for results involving the arithmetical jump operator, but it does not provide a direct control of arithmetical computations. In particular, we do not know how to prove results such as the Splitting or the Density Theorems.

In this section we push the method to a limit by proving the Diamond Theorem, thus providing an elementary difference between $\mathcal{R}_a$ and $\mathcal{R}$.

Minimal pairs

Before getting to the full Diamond Theorem, we concentrate on an intermediate step.

Proposition XIII.5.1 (Simpson [1985]) *There is a minimal pair of ω-r.e.a. a-degrees.*

Proof. We want two ω-r.e.a. sets A_0 and A_1 such that:

- A_0 and A_1 are not arithmetical

- $E \leq_a A_0, A_1 \Rightarrow E$ arithmetical.

By the Posner Lemma (V.2.18.a), we can state the requirements for the second condition as

$$\{e\}^{A_0^{(n)}} \simeq \{e\}^{A_1^{(n)}} \simeq E \;\Rightarrow\; E \text{ arithmetical.}$$

Since we work with ω-hops (as opposed to ω-r.e.a. sets), we actually require a relativized version of this:

$$\{e\}^{A_0(X)^{(n)}} \simeq \{e\}^{A_1(X)^{(n)}} \simeq E \;\Rightarrow\; E \leq_a X. \qquad \text{(XIII.15)}$$

The global problems we face here are similar to those encountered in Section 4. Basically, we want to control the n-th jumps of A_i $(i = 0, 1)$ through a control of A_i itself; and we want to take care of just one condition at a time. We have already found solutions to these problems, by building not a single pair of ω-hops (A_0, A_1), but a double sequence $(A_{0,n}, A_{1,n})_{n\in\omega}$ satisfying

$$A_{i,n}(X)' \equiv_T A_{i,n}(X) \oplus X' \equiv_T A_{i,n+1}(X'). \qquad \text{(XIII.16)}$$

We will use $A_{i,0}$ as A_i. From XIII.16 it follows, by uniformity and induction, that

$$A_{i,0}(X)^{(n)} \equiv_T A_{i,n}(X^{(n)}),$$

and thus we can control $A_{i,n}$ as a way of controlling the n-th jump of $A_{i,0}$. Each level of the sequence will handle a single requirement XIII.15, and we have to decide which one.

We cannot deduce from

$$\begin{gathered}\{e\}^{A_{0,0}(X)^{(n)}} \simeq \{e\}^{A_{1,0}(X)^{(n)}} \simeq E \\ A_{i,0}(X)^{(n)} \equiv_T A_{i,n}(X^{(n)})\end{gathered}$$

that

$$\{e\}^{A_{0,n}(X^{(n)})} \simeq \{e\}^{A_{1,n}(X^{(n)})} \simeq E,$$

since T-reductions are obviously not degree-invariant.[1]

However, there are indices $a^i_{e,n}$ such that

$$\{a^0_{e,n}\}^{A_{0,n}(X^{(n)})} \simeq \{a^1_{e,n}\}^{A_{1,n}(X^{(n)})} \simeq E :$$

[1]For example, if we let

$$\{e\}^Z(x) \simeq \begin{cases} 1 & \text{if } 2x \in Z \\ 0 & \text{otherwise} \end{cases}$$

then $\{e\}^{\emptyset\oplus Y} \simeq \emptyset$, and $\{e\}^{Y\oplus\emptyset} \simeq Y$. Now $Y \oplus \emptyset \equiv_T \emptyset \oplus Y$, but in general $Y \not\equiv_T \emptyset$.

$$\begin{tikzcd}
A_{i,0}(X)^{(n)} \arrow[r] & A_{i,n}(X^{(n)}) \\
\{e\}^{A_{i,0}(X)^{(n)}} \simeq \{a^i_{e,n}\}^{A_{i,n}(X^{(n)})} \arrow[u] \arrow[ur] &
\end{tikzcd}$$

We cannot let the level n itself take care of the requirements relative to $a^i_{e,n}$, because there are still infinitely many such requirements (for different e's). But we could decide, for example, to have level $\langle e, n\rangle$ (which is $\geq n$ by properties of the coding function) take care of the requirement relative to e and n. Since

$$A_{i,0}(X)^{(n)} \leq_T A_{i,0}(X)^{(\langle e,n\rangle)} \equiv_T A_{i,\langle e,n\rangle}(X^{(\langle e,n\rangle)}),$$

as above it follows from

$$\{e\}^{A_{0,0}(X)^{(n)}} \simeq \{e\}^{A_{1,0}(X)^{(n)}} \simeq E$$

that there are indices $b^i_{e,n}$ such that

$$\{b^0_{e,n}\}^{A_{0,\langle e,n\rangle}(X^{(\langle e,n\rangle)})} \simeq \{b^1_{e,n}\}^{A_{1,\langle e,n\rangle}(X^{(\langle e,n\rangle)})} \simeq E :$$

$$\begin{tikzcd}
A_{i,0}(X)^{(\langle e,n\rangle)} \arrow[r] & A_{i,\langle e,n\rangle}(X^{(\langle e,n\rangle)}) \\
A_{i,0}(X)^{(n)} \arrow[u] & \\
\{e\}^{A_{i,0}(X)^{(n)}} \simeq \{b^i_{e,n}\}^{A_{i,\langle e,n\rangle}(X^{(\langle e,n\rangle)})} \arrow[u] \arrow[uur] &
\end{tikzcd}$$

To satisfy XIII.15, it is enough to satisfy

$$\{b^0_{e,n}\}^{A_{0,\langle e,n\rangle}(Z)} \simeq \{b^1_{e,n}\}^{A_{1,\langle e,n\rangle}(Z)} \simeq E \;\Rightarrow\; E \leq_a Z. \qquad \text{(XIII.17)}$$

Indeed, if the premise of XIII.15 holds, then so does the premise of XIII.17 with $Z = X^{(\langle e,n\rangle)}$, and then $E \leq_a X^{(\langle e,n\rangle)}$ and $E \leq_a X$.

Exactly as in Section 4, the determination of $b^i_{e,n}$ from e and n requires previous knowledge of $A_{i,\langle e,n\rangle}$, and thus a use of the Fixed-Point Theorem. The whole construction then rests on a *triple use of the Fixed-Point Theorem*, the other two calls to it being necessary to obtain $A_{i,n}$ from $A_{i,n+1}$, and in the perturbations of the ZBC Lemma.

As usual, we have now reduced the whole construction to a class of similar constructions, of the kind: *given ω-hops D_i, and numbers b_i ($i = 0, 1$), build ω-hops A_i nonarithmetical and such that*

$$A_i(X)' \equiv_T A_i(X) \oplus X' \equiv_T D_i(X')$$
$$\{b_0\}^{A_0(X)} \simeq \{b_1\}^{A_1(X)} \simeq E \;\Rightarrow\; E \leq_a X.$$

We now show how to modify the basic construction to take care of the last requirement. The idea is to have *a single requirement Q with highest priority*, that will try to preserve an inequality of the left-hand side as soon as it sees one. That is, Q will try to enforce

$$\{b_0\}^{A_0(X)}(z) \not\simeq \{b_1\}^{A_1(X)}(z)$$

if possible (as in a typical minimal pair requirement). As in the case of the M_e requirements, Q will have to look-ahead and, while working on one column, will have to approximate some of the next columns to be built later on. As usual, this is achieved by approximations of the kind $A^{[\leq n]}_{i,\tau_i}(X)$, with τ_i a decreasing string.

We will discuss the action of Q as if we were building columns one at a time (instead of in pairs), as we did in Section 3 for the M_e's (the necessary changes being mostly notational).

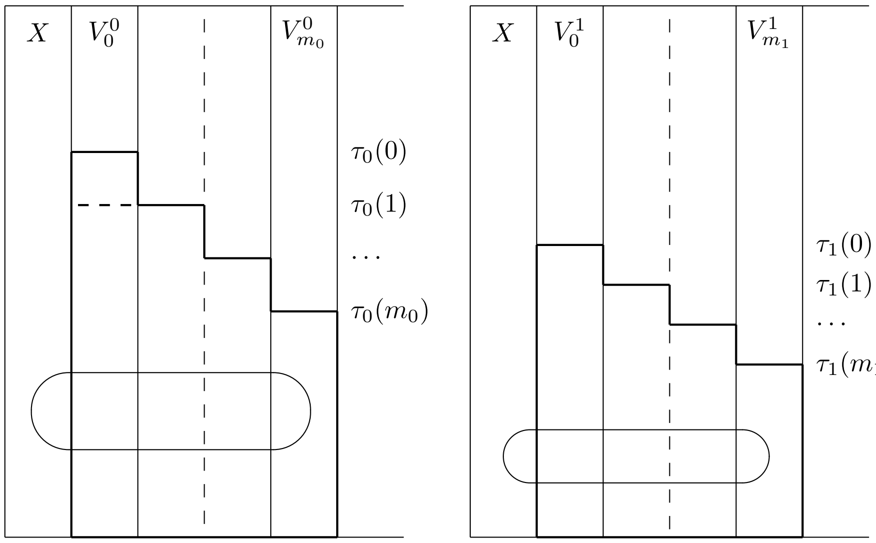
Let us start with the action of Q on V_0. At stage $s + 1$, we will look for $z \leq s$ and τ_0, τ_1 such that

$$s = \tau_0(0) > \tau_0(1) > \cdots > \tau_0(m_0) \tag{XIII.18}$$
$$s = \tau_1(0) > \tau_1(1) > \cdots > \tau_1(m_1) \tag{XIII.19}$$
$$\{b_0\}^{A^{[\leq 0]}_{0,\tau_0}(X)}(z) \not\simeq \{b_1\}^{A^{[\leq 0]}_{1,\tau_1}(X)}(z), \tag{XIII.20}$$

the left-hand and right-hand sides converging, respectively, in at most $\tau_0(m_0)$ and $\tau_1(m_1)$ steps.

We would like to freeze the boldface part of the picture below in both sets, to preserve the converging and disagreeing computations (that we see simultaneously, at stage $s + 1$). We might think of proceeding exactly as we did for the M_e's, by freezing both sets up to $\tau_i(0)$. But we have to give a uniform procedure, that will work for future columns as well, while this one would not. The reason is that for column 0 we can look at both sides simultaneously, since

V_0^i has to be r.e. in X and X is the first column of both A_0 and A_1. But column V_1^i can be r.e. only in X and the previous column V_0^i of the same set we are building, and we cannot refer to or have knowledge of the other side.



So, when building V_1^i, we might independently realize that $\{b_i\}^{A^{[\le 1]}_{i,\tau_i^-}}(z)\downarrow$ (where τ_i^- is τ_i with the first component dropped, i.e. $\langle \tau_i(1), \dots, \tau_i(m_i)\rangle$), but we cannot compare the two computations (since each uses the whole column V_0^i). Then we would not know whether we have to freeze the appropriate amount of V_1^i up to $\tau_i(1)$.

The simplest solution, of freezing whenever we see such a z and such a string, on any side, is too demanding. Indeed, there might be infinitely many such z's, and (not knowing which is the right one producing a disagreement) we would be forced to consider each one as a possible good witness, and we could end up by imposing an infinite restraint (that would spoil the rest of the construction).

We then need cooperation between the two sides (to act only for the right z), yet we cannot have it directly. The oblique way is to have previous columns signal the next one that they have to act. If each column is told to act by the previous one, then the ultimate command originates from column 0, a place in which cooperation between the two sides is directly possible.

Let us go back to column V_1^i, at a stage $s+1$ when Q might act. This means that there are $z \leq s$ and τ_i^- such that

$$s = \tau_i^-(0) > \cdots > \tau_i^-(m_i - 1)$$
$$\{b_i\}^{A^{[\leq 1]}_{i,\tau_i^-}}(z)\downarrow \text{ in at most } \tau_i^-(m_i-1) \text{ steps.}$$

Since action has to be taken at this stage, V_1^i has to know right away whether it has to preserve the computation or not.

The action of Q on V_0^i is thus modified as follows. At stage $\tau_i(0)+1$ (when Q acts), V_0^i is frozen only up to $\tau_i(1)$, instead of up to $\tau_i(0)$, and the $\tau_i(1)+1$ element of the current complement of V_0^i is also enumerated in it.

Then V_1^i is able to know whether it is its turn to act at stage $\tau_i^-(0)+1$. First it runs the construction of V_0^i (recursively in X) until it obtains (recursively in V_0^i, as allowed in the construction of V_1^i) an approximation $V_{0,t}^i$ such that

$$\text{the } \tau_i^-(0)+1\text{-th element of } \overline{V_{0,t}^i} \text{ is the } \tau_i^-(0)+1\text{-th element of } \overline{V_0^i}.$$

After stage t, Q cannot act on V_0^i for any string τ_i such that $\tau_i = \langle \tau_i(0)\rangle * \tau_i^-$, otherwise the $(\tau_i(1)+1) = (\tau_i^-(0)+1)$-th element of $\overline{V_0^i}$ would enter V_0^i. Then V_1^i has only to see whether Q has acted by stage t, for z and a string τ_i as above. If so, Q will act similarly on V_1^i, by freezing the elements up to $\tau_i(2)+1$, and enumerating the $(\tau_i(2)+1)$-th element of the current complement. Similar actions will be taken on the next columns (except that on the last one $V_{m_i}^i$ we just freeze up to $\tau_i(m_i)$, since there is nothing more to signal ahead).

Note that, unlike the M_e's, the requirement Q is not only negative (freezing appropriate parts to preserve computations), but also positive (enumerating one element to signal).

It remains to argue that the construction really works, in the sense that the action taken for Q achieves its goals, and it does not interfere with the rest of the construction.

1. The action described above solves the cooperation problem, but introduces a small variant to the construction. Since the $(\tau_i(1)+1)$-th element of V_0^i can be less that $\tau_i(0)$, although not of $\tau_i(1)$, we preserve V_0^i only up to $\tau_i(1)$, but not necessarily up to $\tau_i(0)$, and similarly in the other columns. We still have to check that, nevertheless,

$$\{b_i\}^{A^{[\leq 0]}_{i,\tau_i}(X)}(z) \simeq \{b_i\}^{A_i(X)}(z),$$

i.e. that Q succeeds by preserving the computation. Recall that, by definition, $A^{[\leq 0]}_{i,\tau_i}(X)$ is the set obtained by:

- running the construction of V_0^i for $\tau_i(0)$ stages over X (this gives the approximation $V^i_{0,\tau_i(0)}$)
- running the construction of V_1^i for $\tau_i(1)$ stages over the approximation of $V^i_{0,\tau_i(0)}$ just obtained, and so on.

Moreover, the computation is supposed to converge in at most $\tau_i(m_i)$ steps. Then, before Q acts we have the following situation:

- the oracle V_0^i can be asked at most up to $\tau_i(m_i)$. So, as long as V_0^i is frozen up to $\tau_i(m_i) \leq \tau_i(1)$, it produces the right answers
- the oracle V_1^i is approximated for $\tau_i(1)$ stages over $V^i_{0,\tau_i(0)}$. Now this approximation can ask the oracle only up to $\tau_i(1)$, but $V^i_{0,\tau_i(0)}$ up to $\tau_i(1)$ is final (by the construction, which freezes V_0^i up to $\tau_i(1)$), and again it produces the right output, and so on.

Thus the action for Q just preserved the right amount, and the computation is final. Hence, if Q acts, then it preserves disagreement.

Note that after Q has acted V_0^i may have changed above $\tau_i(1)$, by the introduction of the $(\tau_i(1)+1)$-th element of $\overline{V_0^i}$ into V_0^i, and thus the argument just given would not be sufficient in general to prove that the restraint we imposed preserves computations from $A^{[\leq 0]}_{i,\tau_i}(X)$. But it does for computations converging before Q acts, which is what we need (since Q precisely intends to preserve one of these).

2. We now have to show that if Q does not act and

$$\{b_0\}^{A_0(X)} \simeq \{b_1\}^{A_1(X)} \simeq E$$

then $E \leq_a X$. In this case, to compute $E(z)$ is enough to find any decreasing string τ_0 such that

$$\{b_0\}^{A^{[\leq 0]}_{0,\tau_0}(X)}(z)\downarrow \text{ in at most } \tau_0(m_0) \text{ steps}$$

(it exists since $\{b_0\}^{A_0(X)}$ is total), with the first column settled down up to $\tau_0(1)$ (to be able to start with the right V_0^0 up to $\tau_0(1)$). This requires knowledge of V_0^0, which is r.e. in X and hence recursive in X'. When such a τ_0 is found, we have the final computation $E(z)$ (since otherwise there would be a similar final computation on the A_1-side, and Q would act and preserve the disagreement). But then $E \leq_T X'$, in particular $E \leq_a X$.

3. A first effect of Q on the rest of the construction is that we really have to work on the first column after X, to have the same situation and be able to cooperate. In particular, we can no longer simply insert V^i_{-1} as a new column between X and V^i_0 (for diagonalization purposes). We will then actually build $V^i_{-1} \oplus V^i_0$ in just one shot. By the freezing effect of Q, we might be unable to code all we want in V^i_{-1}, and finitely many elements might be forced out. Also, by the signalling effect of Q, one unwanted element might enter V^i_{-1}. But this only produces a finite difference with the optimal action we would like to produce, and so it has no effect (as far as the arithmetical degree of $A_0 \oplus A_1$ is concerned).

4. Recall that in Sections 3 and 4 the M_e's had highest priority and were never injured. Now Q has highest priority and is also positive, and it might then injure some M_e. The fact is that Q acts at most once (having highest priority), and its action involves only finitely many columns. Then Q could interfere with only finitely many M_e's, and this is still enough (each M_e is actually considered infinitely many times, due to the fact that each recursive function has infinitely many indices). $\Box$

The Diamond Theorem

With no additional effort we can obtain a stronger result, with interesting consequences.

Theorem XIII.5.2 Diamond Theorem for $\boldsymbol{\mathcal{R}_a}$ (Simpson [1985]) *There are nontrivial ω-r.e.a. a-degrees $\boldsymbol{a}$ and $\boldsymbol{b}$ such that*

$$\boldsymbol{a} \cap \boldsymbol{b} = \mathbf{0}_{\boldsymbol{a}} \quad \textit{and} \quad \boldsymbol{a} \cup \boldsymbol{b} = \mathbf{0}'_{\boldsymbol{a}}.$$

Proof. This is obtained as in XIII.3.5, by an appropriate choice of the columns V^i_{-1}. Note that now $A_0 \oplus A_1 \equiv_a \emptyset^{(\omega)}$, but $A_0 \oplus A_1 \equiv_T \emptyset^{(\omega)}$ does not necessarily hold. The point is that we still have

$$X' \leq_T A_0(X) \oplus A_1(X)$$

but not uniformly so, since the columns V^i_{-1} do code the right sets, but only up to a finite difference. This is, however, enough to produce

$$X^{(\omega)} \leq_a A_0(X) \oplus A_1(X). \quad \Box$$

Corollary XIII.5.3 *For any nontrivial a-degree below $\mathbf{0}'_{\boldsymbol{a}}$ there is an ω-r.e.a. a-degree incomparable with it.*

Proof. Consider the two a-degrees of the Diamond Theorem. Then one of them must be incomparable with the given a-degree, otherwise either they would be comparable (being one above and the other below the given a-degree), or they would not have l.u.b. $\mathbf{0}'_{\boldsymbol{a}}$ (being both below the given a-degree) or g.l.b. $\mathbf{0}_{\boldsymbol{a}}$ (being both above the given a-degree). $\square$

Note that the proof of the corollary is not uniform (we only know that one of two a-degrees of the Diamond Theorem is incomparable with the given one, but do not know which one). We do not know whether the result holds uniformly.

Corollary XIII.5.4 *$\boldsymbol{\mathcal{R}_a}$ and $\boldsymbol{\mathcal{R}}$ are not elementarily equivalent.*

Proof. The Diamond Theorem holds for $\boldsymbol{\mathcal{R}_a}$ by XIII.5.2, but it fails for $\boldsymbol{\mathcal{R}}$ by X.6.23. $\square$

Note that $\boldsymbol{\mathcal{R}_a}$ *is not elementarily equivalent to* $\boldsymbol{\mathcal{R}_{wtt}}$ *and* $\boldsymbol{\mathcal{R}_m}$ either, again by the Diamond Theorem. No elementary difference with $\boldsymbol{\mathcal{R}_{btt}}$ and $\boldsymbol{\mathcal{R}_{tt}}$ is known, although the nonexistence of minimal ω-r.e.a. a-degrees would provide one.

æ

Chapter XIV

Enumeration Degrees

Chapters IX and X provided a microscopic analysis of the recursively enumerable sets. We now proceed to study the continuum modulo the recursively enumerable sets by means of **enumeration reducibility**, which was introduced in Section II.3 as a possible reducibility for partial functions. Here, as in the rest of the book, we are concerned with its version for sets, which we define in Section 1 and study in Section 2. Section 3 deals with the theory of e-degrees of $\mathbf{\Sigma_2^0}$ **sets**, which provide an analogue of the r.e. sets in the context of enumeration reducibility.

The connections with relative computability of partial functions are spelled out in Section 1, while Section 4 shows how the functionals involved in enumeration reducibility provide a countable effective version of the graph model for the λ-calculus.

XIV.1 Enumeration Degrees

Enumeration reducibility was introduced in Section II.3 as a tool for comparing partial functions, but it also presents itself naturally in the study of sets.

Definition XIV.1.1 (Friedberg and Rogers [1959]) *A is* ***e*-reducible** *to B ($A \leq_e B$) if, for some recursively enumerable relation R,*

$$x \in A \Leftrightarrow (\exists u)(R(x,u) \wedge D_u \subseteq B).$$

A is ***e*-equivalent** *to B ($A \equiv_e B$) if $A \leq_e B$ and $B \leq_e A$.*

The intuition behind the definition of $A \leq_e B$ is that we can effectively enumerate A whenever we are given any enumeration of B. This is stronger than simply saying that A is r.e. in B, since in the latter case we can enumerate A using any kind of information from B (including negative information from its complement) while in the former we are only allowed positive information.

Exercises XIV.1.2 (Selman [1971])

a) *$A \leq_e B$ if and only if, for every X, A is r.e. in X whenever B is r.e. in X.* (Hint: if $A \not\leq_e B$, we build X such that B is r.e. in X but A is not. To make B r.e. in X, we let

$$x \in B \Leftrightarrow (\exists y)(\langle x, y\rangle \in X).$$

To make A not r.e. in X, we satisfy the following requirements:

$$R_e \quad : \quad A \neq \mathcal{W}_e^X.$$

We build X by finite extensions σ_s, starting with $\sigma_0 = \emptyset$. At stage $e+1$ we are given σ_e and we satisfy R_e as follows. We would like to consider only strings $\sigma \supseteq \sigma_e$ such that

$$x \in B \Leftrightarrow (\exists y)(\sigma(\langle x, y\rangle) = 1),$$

but the left-to-right implication uses negative information from B. So we call a string $\sigma \supseteq \sigma_e$ *admissible* if it satisfies the right-to-left implication for all x, and the left-to-right implication for all $x \leq e$. There are two cases:

1. If $(\exists x)(\exists \sigma \text{ admissible})(x \in \mathcal{W}_e^\sigma - A)$, let $\sigma_{e+1} = \sigma$ for one such σ. Then $x \in \mathcal{W}_e^X - A$ and R_e is satisfied.
2. If $(\exists x)(\forall \sigma \text{ admissible})(x \in A - \mathcal{W}_e^\sigma)$, let $\sigma_{e+1} = \sigma_e$. Then $x \in A - \mathcal{W}_e^X$, since X will be defined in the future only using admissible strings, and R_e is satisfied.

Suppose none of the two cases applies. Then

$$x \in A \Leftrightarrow (\exists \sigma \text{ admissible})(x \in \mathcal{W}_e^\sigma),$$

the right-to-left implication because case 1 fails and the left-to-right one because case 2 fails. Since the condition of admissibility uses arbitrary positive information from B, but only a fixed finite negative information from it, it follows that $A \leq_e B$, contradiction.)

b) *$\leq_e$ is maximal among the subreducibilities of 'being r.e. in', i.e. among the reducibilities $\leq_r$ such that*

$$A \leq_r B \wedge B \in \Sigma_1^{0,X} \Rightarrow A \in \Sigma_1^{0,X}.$$

(Hint: if $A \leq_r B$, then $A \leq_e B$ by part a).)

By the S_n^m-Theorem II.1.7, an equivalent condition for $A \leq_e B$ is the existence of a recursive function f such that

$$x \in A \Leftrightarrow (\exists u)(u \in \mathcal{W}_{f(x)} \wedge D_u \subseteq B).$$

In particular, e-reducibility is a natural generalization of positive reducibility, which is defined in the same way, only with finite sets $D_{f(x)}$ in place of r.e. sets $\mathcal{W}_{f(x)}$ (see Section III.3).

Exercises XIV.1.3 a) *If A is r.e., then $A \leq_e B$ for any set B.*

b) *If $A \leq_e B$ and B is r.e., then so is A.*

c) *If $A \leq_p B$, then $A \leq_e B$.*

d) *There are sets A such that $A \not\leq_e \overline{A}$. Thus btt-reducibility does not imply e-reducibility.* (Hint: by part b), letting $A = \overline{\mathcal{K}}$.)

The reductions used in e-reducibility (called **enumeration operators** or, briefly, ***e*-operators**) play a role similar to that of recursive functionals in the study of Turing reducibility. We thus introduce for the former a notation reminiscent of the latter.

Definition XIV.1.4

1. *$\boldsymbol{\Phi_e}$ is the e-th enumeration operator:*

$$\Phi_e(A) = \{x : (\exists u)(\langle x, u\rangle \in \mathcal{W}_e \ \wedge \ D_u \subseteq A)\}$$

2. *$\boldsymbol{\Phi_{e,s}}$ is the finite approximation of Φ_e of level s:*

$$\Phi_{e,s}(A) = \{x : (\exists u)(\langle x, u\rangle \in \mathcal{W}_{e,s} \ \wedge \ D_u \subseteq A)\}$$

Obviously, $\Phi_e(A)$ is r.e. in A and $\Phi_{e,s}(A)$ is finite and uniformly recursive in s and A (because $\mathcal{W}_{e,s}$ is finite and uniformly recursive in s).

Note that $\leq_e$ is a reflexive and transitive relation. Thus $\equiv_e$ is an equivalence relation.

Definition XIV.1.5 *The equivalence classes of sets w.r.t. e-equivalence are called* ***e*-degrees** *or* **partial degrees** *and $(\boldsymbol{\mathcal{D}_e}, \leq)$ is the structure of e-degrees, with the partial ordering $\leq$ induced on them by $\leq_e$.*

$\mathbf{0_e}$ *is the least e-degree, containing exactly the r.e. sets.*

Total degrees

Before starting an examination of the structure $\boldsymbol{\mathcal{D}_e}$ we investigate an interesting class of e-degrees, obtained by looking at graphs G_α of partial functions α.

Notice that $\alpha \leq_e \beta$ holds according to the definition of e-reducibility for partial functions given on p. I.198 if and only if $G_\alpha \leq_e G_\beta$ holds according to the definition XIV.1.1 for sets.

Exercise XIV.1.6 *Every e-degree contains the graph of a partial function.* (Hint: given A, consider $\varphi(x) \simeq 0 \Leftrightarrow x \in A$.)

Definition XIV.1.7 *An e-degree is called* **total** *if it contains the graph of a total function.*

The next proposition provides sufficient conditions for the e-degree of A to be total.

Proposition XIV.1.8 *The following are equivalent, for any set A:*

1. $A \equiv_e c_A$
2. $A \equiv_e A \oplus \overline{A}$
3. $\overline{A} \leq_e A$.

Proof. 1 and 2 are equivalent because $c_A \equiv_e A \oplus \overline{A}$. Indeed, any enumeration of the graph of c_A produces an enumeration of both A and $\overline{A}$: x is enumerated in A or $\overline{A}$, according to whether $\langle x, 1\rangle$ or $\langle x, 0\rangle$ is enumerated in the graph of c_A. And any enumeration of $A \oplus \overline{A}$, produces an enumeration of both A and $\overline{A}$, and hence of c_A, as follows: $\langle x, 1\rangle$ or $\langle x, 0\rangle$ is enumerated in the graph of c_A, according to whether x is enumerated in A or in $\overline{A}$.

2 obviously implies 3. And 3 implies 2 because if $\overline{A} \leq_e A$, then $A \oplus \overline{A} \leq_e A$, while the converse always holds. $\Box$

We now show that the previous conditions are not only sufficient but also necessary, in the sense that any total e-degree contains a set A satisfying them.

Corollary XIV.1.9 *An e-degree is total if and only if it contains the graph of a characteristic function.*

Proof. Notice that if f is total, then $\overline{G}_f \leq_e G_f$ because

$$\langle x, z\rangle \notin G_f \;\Leftrightarrow\; (\exists y)(y \neq z \,\wedge\, \langle x, y\rangle \in G_f).$$

It follows from the proposition that $G_f \equiv_e c_{G_f}$, i.e. the graph of a total function is e-equivalent to its own characteristic function. $\Box$

Since a nontotal e-degree cannot contain characteristic functions, the existence of nontotal e-degrees (XIV.1.11) shows that we cannot identify, from the point of view of e-reducibility, a set and its characteristic function.

Exercises XIV.1.10 a) *An e-degree is total if and only if it contains a retraceable (or a regressive) set.* (Case [1971]) (Hint: given a total function f, consider the set A of the sequence numbers $\hat{f}(n) = \langle f(0), \ldots, f(n)\rangle$. A is retraceable via the function that chops off the last component of any sequence number. And $A \equiv_e G_f$, because from any enumeration of one we can obtain an enumeration of the other. Thus every total e-degree contains a retraceable (and hence a regressive and an introreducible) set.

If A is regressive, from any enumeration of it we can obtain the canonical enumeration induced by the regressive function. If f describes such an enumeration, i.e. if $f(n)$ is its n-th element, we have $G_f \leq_e A$. Obviously $A \leq_e G_f$, since from any enumeration of f we obtain an enumeration of A. Thus $A \equiv_e G_f$ and the e- degree of A is total.)

b) *An e-degree is total if and only if it contains an introreducible set.* (Jockusch) (Hint: one direction follows from part a), since every retraceable set is introreducible by II.6.7.a. Conversely, given A infinite and introreducible, we prove $\overline{A} \leq_e A$ using XIV.1.2. Suppose A is r.e. in X and let C be an infinite subset of A recursive in X, by a relativization of II.1.20. Since $A \leq_T C$ by introreducibility, A is recursive in X by transitivity. Hence $\overline{A}$ is r.e. in X by a relativization of II.1.19.)

c) *An e-degree is total if and only if it contains a Turing jump.* (Case [1974]) (Hint: by XIV.1.9, it is enough to prove that $A' \equiv_e c_A$. Since $A, \overline{A} \leq_1 A'$, we have $c_A \leq_e A'$. Since A' is r.e. in A, $A' \leq_e A \oplus \overline{A}$ and hence $A' \leq_e c_A$.)

The next result shows that the notion of total e-degree is not trivial.

Theorem XIV.1.11 (Medvedev [1955a]) *There are nontotal e-degrees.*

Proof. It is enough to build a set A which is not r.e. and has no total predecessors (such a set is called **quasi-minimal**). The conditions to be satisfied are the following:

$$\begin{array}{lll} R_{2e} & : & A \neq \mathcal{W}_e \\ R_{2e+1} & : & \Phi_e(A) \text{ graph of a total function } \Rightarrow \Phi_e(A) \text{ r.e.} \end{array}$$

We build A by finite initial segments, starting with $\sigma_0 = \emptyset$. At stage $s+1$, let σ_s be given.

- If $s = 2e$, then we satisfy R_{2e} by choosing the least x such that $\sigma_s(x)$ is not yet defined and extending σ_s to σ_{s+1} by letting
$$\sigma_{s+1}(x) = 1 - \mathcal{W}_e(x).$$
- If $s = 2e+1$, then we try to vacuously satisfy R_{2e+1} by forcing $\Phi_e(A)$ not to be single-valued. We see if there is an initial segment $\sigma \supseteq \sigma_s$ such that, for some x, y and z,
$$y \neq z \ \wedge \ \langle x,y\rangle, \langle x,z\rangle \in \Phi_e(\sigma^+),$$
where σ^+ is the set determined by σ, i.e. $\{x : \sigma(x) \simeq 1\}$. If so, choose such a σ and let $\sigma_{s+1} = \sigma$. Otherwise let $\sigma_{s+1} = \sigma_s$.

We have to argue that this strategy satisfies R_{2e+1}. Suppose $\Phi_e(A)$ is the graph of a total function; we show that (for $s = 2e+1$)

$$\langle x,y\rangle \in \Phi_e(A) \quad \Leftrightarrow \quad (\exists \sigma \supseteq \sigma_s)[\langle x,y\rangle \in \Phi_e(\sigma^+)], \qquad \text{(XIV.1)}$$

so that $\Phi_e(A)$ is r.e.

The left-to-right implication is obvious. For the converse, suppose the value y is wrong. Then some other string $\tau \supseteq \sigma_s$ with $\tau^+ \subseteq A$ would give the right value $z \neq y$ (because $\Phi_e(A)$ is total by hypothesis). Then we would have

$$\langle x, y\rangle \in \Phi_e(\sigma^+) \ \wedge\ \langle x, z\rangle \in \Phi_e(\tau^+).$$

But then any string extending σ_s and determining a set containing $\sigma^+ \cup \tau^+$ would give two values (note that only the positive information is needed), and the construction at step $s+1$ would ensure that $\Phi_e(A)$ is not the graph of a total function. $\Box$

Note that if we only suppose that $\Phi_e(A)$ is the graph of a partial function, then y might be wrong simply because there is *no* value, and we would not be able to proceed towards a contradiction. In other words, the construction does not automatically produce a minimal e-degree (with good reason, since by XIV.2.9 none exists). What the proof does show is that if $\Phi_e(A)$ is the graph of a partial function, then it has a partial recursive extension (defined by the right-hand-side of XIV.1 above).

Since the proof of the result is by finite extensions, the usual considerations (see V.3.13) show that *the set of nontotal e-degrees is comeager* (Myhill [1961a]), so that almost every e-degree is nontotal. Moreover, *the total e-degrees generate* $\boldsymbol{\mathcal{D}_e}$ (see XIV.2.5.c) and thus provide an automorphism base for it. However, *the total e-degrees are not invariant under automorphisms of* $\boldsymbol{\mathcal{D}_e}$ and thus they are not definable in $\boldsymbol{\mathcal{D}_e}$ (Cooper [199?c]).

Exercises XIV.1.12 The jump operator. (Selman [1971], Cooper [1984], McEvoy [1985]) Let $J(A) = K_A \oplus \overline{K_A}$, where $K_A = \{x : x \in \Phi_x(A)\}$ is the natural diagonal set in the context of e-reducibility. The e-jump $\boldsymbol{a'}$ of an e-degree $\boldsymbol{a}$ is the e-degree of $J(A)$, for any $A \in \boldsymbol{a}$. The reason to use both K_A and $\overline{K_A}$ is to make the e-jump agree with the Turing jump on the total e-degrees, see part d) below.

a) *The e-jump is well-defined on e-degrees.* (Hint: as in V.1.6, one can prove $A \leq_e B \Leftrightarrow K_A \leq_1 K_B$. It follows that if $A \leq_e B$, then $J(A) \leq_1 J(B)$.)

b) $A <_e J(A)$. (Hint: by diagonalization.)

c) $J(\emptyset) \equiv_e \overline{\mathcal{K}}$. (Hint: notice that $K_\emptyset \equiv_1 \mathcal{K}$, and $\mathcal{K} \oplus \overline{\mathcal{K}} \equiv_e \overline{\mathcal{K}}$ because $\mathcal{K}$ is r.e.)

d) $J(c_A) \equiv_e c_{A'}$. *In particular, the e-jump agrees with the Turing jump on the total e-degrees.* (Hint: notice that $A \oplus \overline{A} \equiv_e c_A$. Moreover, $K_{A\oplus\overline{A}} \equiv_e A'$ because K_X and X' are, respectively, sets Σ^0_1-complete w.r.t. to positive and arbitrary, i.e. positive and negative, information about X.)

e) *The range of the e-jump is the set of total e-degrees above* $\boldsymbol{0'_e}$. (Hint: by part d) and the Jump Inversion Theorem V.2.24 for Turing degrees.)

XIV.2 The Theory of Enumeration Degrees

This section is devoted to a study of the structure $\boldsymbol{\mathcal{D}_e}$ introduced above. We proceed along the lines of Chapter V and use the same conventions and notation.

We start with an analogue of V.1.12.

Proposition XIV.2.1 (Case [1971], Selman [1971]) *As a partially ordered structure, $\boldsymbol{\mathcal{D}_e}$ is an uppersemilattice of cardinality $2^{\aleph_0}$ that is not a lattice and has a least element, but no maximal one. Moreover, each element has $2^{\aleph_0}$ successors and at most countably many predecessors.*

Proof. As in V.1.12, by noting that $\mathbf{0_e}$ is the least e-degree, $A <_e J(A)$ (by XIV.1.12.b) and $A \oplus B$ is the least upper bound of A and B w.r.t. e-reducibility.

It remains to show that $\boldsymbol{\mathcal{D}_e}$ is not a lattice. Let A be a Σ^1_1-complete set. If $B \leq_e A, \overline{A}$, then B is r.e. in both A and $\overline{A}$, hence is Δ^1_1 and so must be B''. Then $B'' \leq_1 A, \overline{A}$ by Σ^1_1-completeness of A and hence $B'' \leq_e A, \overline{A}$. Since $B <_e B''$, B is not the g.l.b. of A and $\overline{A}$. $\square$

The following is a simple but useful observation.

Proposition XIV.2.2 (Myhill [1961a]) *$\boldsymbol{\mathcal{D}}$ is isomorphic to the structure of the total e-degrees. Moreover, the isomorphism preserves least element, l.u.b.'s and jumps.*

Proof. Consider the function $t : \boldsymbol{\mathcal{D}} \longmapsto \boldsymbol{\mathcal{D}_e}$ defined by

$$t(\boldsymbol{a}) = \text{the } e\text{-degree of } c_A, \text{ for any } A \in \boldsymbol{a}.$$

By II.3.1 and II.3.25.3,

$$A \leq_T B \ \Leftrightarrow\ c_A \leq_T c_B \ \Leftrightarrow\ c_A \leq_e c_B,$$

and in particular

$$A \equiv_T B \ \Leftrightarrow\ c_A \equiv_e c_B.$$

It follows that t is well-defined on the T-degrees, it preserves the order and it is one-one. By XIV.1.9, t is onto the total e-degrees. Thus t is an isomorphism of the T-degrees and the total e-degrees.

Since $c_\emptyset$ is recursive, t sends $\mathbf{0}_T$ into $\mathbf{0}_e$ and thus it preserves the least element. By XIV.1.12.d, t preserves the jump. Finally, t preserves the l.u.b. because $c_{A \oplus B} \equiv_e c_A \oplus c_B$. $\square$

The previous isomorphism automatically provides a good deal of information about $\boldsymbol{\mathcal{D}_e}$. For example, any countable partial ordering is embeddable in $\boldsymbol{\mathcal{D}}$ (V.2.9) and hence in the total e-degrees. In particular, *the one-quantifier theory of $\boldsymbol{\mathcal{D}_e}$ is decidable* (V.2.10).

Minimal pairs

In many cases when one cannot directly translate results from $\boldsymbol{\mathcal{D}}$ to $\boldsymbol{\mathcal{D}_e}$ as above, one can at least adapt their proofs and thus indirectly translate them. As an illustration, we provide the analogue of V.2.16 by adapting the finite extensions method to e-reductions.

Proposition XIV.2.3 (Case [1971], Jockusch) *There exists a minimal pair of e-degrees. Actually, each nonzero e-degree is part of a minimal pair.*

Proof. Let B be a non-r.e. set. We want A such that

$$\begin{array}{lcl} R_{2e} & : & A \neq \mathcal{W}_e \\ R_{2\langle e,i\rangle+1} & : & \Phi_e(A) = \Phi_i(B) = C \ \Rightarrow \ C \text{ r.e.} \end{array}$$

We build A by finite initial segments, starting with $\sigma_0 = \emptyset$. At stage $s+1$, let σ_s be given.

1. If $s = 3e$, then we satisfy R_{2e} by choosing the least x such that $\sigma_s(x)$ is not yet defined, and extending σ_s to a σ_{s+1} such that
$$\sigma_{s+1}(x) = 1 - \mathcal{W}_e(x).$$
2. If $s = 3\langle e,i\rangle + 1$, then we try to vacuously satisfy $R_{2\langle e,i\rangle+1}$ by forcing an inequality of the two sides. We see if there is x and $\sigma_0, \sigma_1 \supseteq \sigma_s$ such that Φ_e gives two different answers on x, i.e.
$$x \in \Phi_e(\sigma_0^+) \ \wedge \ (\forall \sigma \supseteq \sigma_1)[x \notin \Phi_e(\sigma^+)],$$
where σ^+ is the set determined by σ. If so, choose as σ_{s+1} the one between σ_0 and σ_1 that gives a different answer from $\Phi_i(B)$, i.e.
$$\sigma_{s+1} = \begin{cases} \sigma_0 & \text{if } x \notin \Phi_i(B) \\ \sigma_1 & \text{if } x \in \Phi_i(B). \end{cases}$$
Otherwise, let $\sigma_{s+1} = \sigma_s$.
3. If $s = 3\langle e,x\rangle + 2$, then we ensure that $x \in \Phi_e(A)$ if possible. We see if there is $\sigma \supseteq \sigma_s$ such that $x \in \Phi_e(\sigma^+)$. If so, we choose one and let $\sigma_{s+1} = \sigma$. Otherwise, we let $\sigma_{s+1} = \sigma_s$.

We have to argue that $R_{2\langle e,i\rangle+1}$ is satisfied. If $\Phi_e(A) = \Phi_i(B)$, we show that (for $s = 3\langle e,i\rangle + 1$)

$$x \in \Phi_e(A) \quad \Leftrightarrow \quad (\exists \sigma \supseteq \sigma_s)[x \in \Phi_e(\sigma^+)], \qquad \text{(XIV.2)}$$

so that $\Phi_e(A)$ is r.e.

The left-to-right implication is obvious. For the converse, suppose there is $\sigma \supseteq \sigma_s$ such that $x \in \Phi_e(\sigma^+)$. Since $\Phi_e(A) = \Phi_i(B)$, the case hypothesis at stage $s+1$, i.e. part 2 above, cannot be satisfied, otherwise we would make $\Phi_e(A) \neq \Phi_i(B)$. Then

$$(\forall \sigma_1 \supseteq \sigma_s)(\exists \sigma \supseteq \sigma_1)(x \in \Phi_e(\sigma^+)).$$

In particular at stage $3\langle x, e\rangle + 3$ we can ensure, if it is not yet the case, that $x \in \Phi_e(A)$. $\square$

Notice the similarities with the proof of XIV.1.11. Both arguments work because of a certain maximality of computations, expressed by the right-to-left implications of XIV.1 and XIV.2, and ensured by the hypothesis of totality in the first case and by a direct action taken in the third part of the construction in the last case.

Notice also the differences with the Turing degrees. First, we cannot use the Posner Lemma (V.2.18) to replace the requirements $R_{\langle e,i\rangle+1}$ by the simpler ones

$$R_{2e+1} \quad : \quad \Phi_e(A) = \Phi_e(B) = C \ \Rightarrow \ C \text{ r.e.}$$

Indeed, the proof of V.2.18 makes use of a T-reduction which is not an e-reduction, because it relies on negative information about the oracle.

Second, and more substantially, *relativization* is not trivial in the context of e-degrees: it usually works for total e-degrees, but not in general. Such is the case, for example, of the proof that there is no minimal e-degree (XIV.2.9): while no total e-degree has minimal covers (XIV.2.10.a), there are e-degrees that do (Cooper [1990a]).

In the previous proof of the existence of minimal pairs the difficulty for relativization lies in part 3 of the construction, which apparently conflicts with coding. By a slightly different construction that avoids this part we now show that, in this special case, the result does fully relativize.

Proposition XIV.2.4 (Rozinas [1978]) *Every e-degree is branching.*

Proof. Let $D <_e B$. We want A such that

$$\begin{array}{lcl} R_{2e} & : & A \neq \Phi_e(D) \\ R_{2\langle e,i\rangle+1} & : & \Phi_e(A \oplus D) = \Phi_i(B) = C \ \Rightarrow \ C \leq_e D. \end{array}$$

The requirements R_{2e} force $D <_e A \oplus D$, while the requirements R_{2e+1} make $A \oplus D$ and B a pair with g.l.b. D.

We build A by finite initial segments, starting with $\sigma_0 = \emptyset$. At stage $s+1$, let σ_s be given.

1. If $s = 2e$, then we satisfy R_{2e} by choosing the least x such that $\sigma_s(x)$ is not yet defined, and extending σ_s to a σ_{s+1} such that
$$\sigma_{s+1}(x) = 1 - \Phi_e(D)(x).$$

2. If $s = 2\langle e, i\rangle + 1$, then we try to vacuously satisfy $R_{2\langle e,i\rangle+1}$ by forcing an inequality of the two sides. We see if there is x and $\sigma \supseteq \sigma_s$ such that
$$x \in \Phi_e(\sigma^+ \oplus D) \ \wedge \ x \notin \Phi_i(B),$$
where σ^+ is the set determined by σ. If so, choose as σ_{s+1} one such σ. Otherwise, let $\sigma_{s+1} = \sigma_s$.

We have to argue that $R_{2\langle e,i\rangle+1}$ is satisfied. If $\Phi_e(A\oplus D) = \Phi_i(B)$, we show that (for $s = 2\langle e, i\rangle + 1$)
$$x \in \Phi_e(A \oplus D) \quad \Leftrightarrow \quad (\exists \sigma \supseteq \sigma_s)[x \in \Phi_e(\sigma^+ \oplus D)], \tag{XIV.3}$$
so that $\Phi_e(A \oplus D) \leq_e D$.

The left-to-right implication is obvious. For the converse, suppose there is $\sigma \supseteq \sigma_s$ such that $x \in \Phi_e(\sigma^+ \oplus D)$. If $x \notin \Phi_e(A \oplus D)$, then $x \notin \Phi_i(B)$ by the hypothesis $\Phi_e(A\oplus D) = \Phi_i(B)$, and the construction at stage $2\langle e, i\rangle + 1$ would ensure that $\Phi_e(A \oplus D) \neq \Phi_i(B)$, contradiction. $\square$

The next exercises provide the analogues of V.4.5 and V.4.8, by adapting the coinfinite extensions method to e-reductions.

Exercises XIV.2.5 a) *Every countable ideal of e-degrees has an exact pair.* (Case [1971], McEvoy and Cooper [1985]) (Hint: extend the proof of V.4.3 as in XIV.2.4.)

b) $\boldsymbol{\mathcal{D}_e}$ *is not a lattice.* (Case [1971], Selman [1971]) (Hint: by part a), applied to any ascending sequence of e-degrees. See XIV.2.1 for a different proof.)

c) *Every e-degree is the g.l.b. of two total e-degrees.* (Rozinas [1978]) (Hint: extend the proof of XIV.2.4 by making the e-degree of $A\oplus D$ total. Then choose the e-degree of B total.)

Minimal covers

Having seen how to adapt the finite and the coinfinite extension methods to the study of $\boldsymbol{\mathcal{D}_e}$, one should turn to the tree method for the construction of minimal e-degrees. But a moment of reflection shows that a totality assumption for the reductions involved was essential in the proofs of the relevant lemmas in Chapter V (in particular, of the crucial V.5.9).

Exercises XIV.2.6 A nonzero e-degree is **minimal-like** if every partial function of strictly smaller e-degree has a partial recursive extension.

a) *There is a nontotal minimal-like e-degree.* (Medvedev [1955a]) (Hint: see XIV.1.11 and the comments following it.)

b) *There is a total minimal-like e-degree.* (Copestake [1988]) (Hint: use the analogue of the construction of minimal Turing degrees. Say that σ e-splits if it has extensions σ_1 and σ_2 such that, for some x, y and z with $y \neq z$,

$$\langle x, y\rangle \in \Phi_e(\sigma_1) \ \text{ and } \ \langle x, z\rangle \in \Phi_e(\sigma_2).$$

The argument of V.2.9 shows that, for a recursive tree T and A on T, if T is e-splitting, then $c_A \leq_e \Phi_e(c_A)$, and if T has no e-splitting, then $\Phi_e(c_A)$ has a partial recursive extension, whenever it is the graph of a partial function.)

In fact, not only does the tree method not help in the construction of minimal e-degrees, but nothing works. The remainder of this subsection is devoted to a proof that minimal e-degrees do not exist, and thus that there is no analogue for $\boldsymbol{\mathcal{D}_e}$ of the results of Sections V.4 and V.5 for $\boldsymbol{\mathcal{D}}$.

Recall that A has minimal e-degree if:

- A is not r.e.
- for every e, either $\Phi_e(A)$ is r.e. or $A \equiv_e \Phi_e(A)$.

The next result imposes a strong limitation on possible minimal e-degrees.

Proposition XIV.2.7 (Gutteridge [1971]) *If A has a minimal e-degree, then $A \in \Delta^0_2$.*

Proof. We show that there is an e-operator Ψ such that, for every A,

$$\Psi(A) \text{ r.e. or } A \equiv_e \Psi(A) \quad \Rightarrow \quad A \in \Delta^0_2. \tag{XIV.4}$$

Note that finding an e-operator satisfying only one of the two conditions is a trivial matter. If $\Psi(A)$ does not depend on A, then it is r.e. for any A, and hence if $A \equiv_e \Psi(A)$, then A is r.e. If $\Psi(A) = A$, then A is r.e. when $\Psi(A)$ is r.e. The only finite sets $D \subseteq A$ that we need to consider in these examples are $\emptyset$ in the first case and singletons in the second.

We will now combine the two approaches and define an r.e. set B on ω seen as a set of pairs, with the following properties; we will put in B a finite initial segment of each column, so that for each x there will be a number n_x such that

$$B = \{\langle x, n\rangle : n \leq n_x\}.$$

For a given A, we will let

$$\Psi(A) = \{\langle x, n\rangle : n < n_x\} \cup \{\langle x, n_x\rangle : x \in A\}.$$

Thus $\Psi(A)$ consists of a fixed part independent of A (the interior of B, i.e. all columns of B except their last elements) plus a part coding A (the image of A on the boundary of B, i.e. the last elements of the columns of B indexed by elements of A).

We will construct B in stages as an r.e. set. The approximation B_s will produce an approximation Ψ_s of Ψ, defined in the same way as Ψ (with B_s in place of B). By definition, $\Psi(A) = \bigcup_{s\in\omega} \Psi_s(A)$. Thus Ψ is an enumeration operator.

The definition of Ψ already ensures that

$$\Psi(A) \text{ r.e.} \;\Rightarrow\; A \in \Delta_2^0. \tag{XIV.5}$$

Indeed, by construction,

$$x \in A \;\Leftrightarrow\; \text{the last element of the } x\text{-th column of } B \text{ is in } \Psi(A).$$

The last element of the x-th column of B can be obtained recursively in $\mathcal{K}$ because B is r.e. (check whether $\langle x, n\rangle \in B$, until the first $n+1$ for which this fails is found). When $\Psi(A)$ is r.e., then also the final question of whether this element is in $\Psi(A)$ can be answered recursively in $\mathcal{K}$.

Since $\Psi(A) \le_e A$ always holds, where Ψ is an enumeration operator, it now remains to define B in such a way as to ensure that

$$A \le_e \Psi(A) \;\Rightarrow\; A \in \Delta_2^0. \tag{XIV.6}$$

If $A \le_e \Psi(A)$, then $A = \Phi_e\Psi(A)$ for some e; this means that we can answer questions of the form $x \in A$ by an r.e. procedure (depending on Φ_e and the fixed definition of Ψ) that uses finite positive information D on A. The idea is to force D to contain only elements smaller than x, so that A can be built inductively using the r.e. procedure and the previous knowledge of A itself, thus forcing A to be r.e. (and in particular Δ_2^0).

We start with $B_0 = \emptyset$. At stage $s+1$, let B_s be given (consisting of finite initial segments of the columns with index $< s$). First, we put $\langle s, 0\rangle$ in B_s, so that the s-th column is not empty (and at the end no column will be). Then for each $e \le x \le s$ (to keep the construction recursive) see if there is a finite set D_u such that

$$\langle x, u\rangle \in \mathcal{W}_{e,s} \;\wedge\; D_u \subseteq B_s.$$

If so, choose the one with least index. We ensure that the elements of D_u on the columns with index $\ge x$ are not the last ones in the columns of B, so that for any A we can decide whether $D_u \subseteq \Psi(A)$ solely on the basis of the columns with index $< x$. To ensure this, it is enough to put one more element on top of all columns of B of index $\ge x$.

By construction, B is r.e. and contains only finitely many elements of each column. Indeed, on column n we put one element to start with and add one element only for the sake of some finite set D_u relative to $e \leq x \leq n$, of which there are only finitely many (for each e and x we consider only finite sets with a minimal index; once one such set has been considered, only ones with smaller index will possibly be considered in the following).

Suppose now $A = \Phi_e\Psi(A)$. Then

$$x \in A \ \Leftrightarrow \ (\exists u)[\langle x, u\rangle \in \mathcal{W}_e \wedge D_u \subseteq \Psi(A)].$$

If $x \geq e$ is in A, then, by definition of Ψ, there are a minimal u and an s such that

$$\langle x, u\rangle \in \mathcal{W}_{e,s} \wedge D_u \subseteq B_s.$$

The construction ensures that all columns of D_u with index $\geq x$ are in $\Psi(A)$. Thus we can decide whether $D_u \subseteq \Psi(A)$ using only information from A up to x. Then A can be effectively generated (given the first e elements) and A is r.e. $\Box$

Exercise XIV.2.8 *An e-degree can have at most countably many minimal covers.* (Gutteridge [1971]) (Hint: relativize the proof above to show that if the e-degree of A is a minimal cover of the e-degree of C, then $A \leq_T C'$.)

By the previous result, only Δ^0_2 sets can have a minimal e-degree. The next result rules out this case, too.

Theorem XIV.2.9 (Gutteridge [1971]) *There is no minimal e-degree.*

Proof. We show that if $A \in \Delta^0_2$ is not r.e., then there is an enumeration operator Ψ such that $\emptyset <_e \Psi(A) <_e A$.

We define an r.e. set B on ω seen as a set of pairs, with the following properties: B will consist of two parts B_0 and B_1 and we will let

$$\Psi(A) = B_0 \cup \{\langle x, n\rangle : \langle x, n\rangle \in B_1 \wedge n \in A\}.$$

Thus $\Psi(A)$ consists of a fixed part B_0 independent of A, plus partial codings of A on each column (of course A will be successfully coded on a given column only if the whole column is contained in B and at most a finite part of it is contained in B_0).

We will construct B by stages as an r.e. set. The approximation B_s will produce an approximation Ψ_s of Ψ, defined in the same way as Ψ (with B_s in place of B). By definition, $\Psi(A) = \bigcup_{s\in\omega} \Psi_s(A)$. Thus Ψ is an enumeration operator and $\emptyset \leq_e \Psi(A) \leq_e A$ for any A.

To ensure that B is r.e., we need a recursive construction. Since $A \in \Delta_2^0$, by the Limit Lemma (IV.1.17) we can use a recursive approximation $\{A_s\}_{s\in\omega}$ to A, consisting of finite sets and correct in the limit. In particular, by definition of Ψ, the limit of $\Psi_s(A_s)(y)$ exists for every y (this property is not true in general and holds only because of the specific form of Ψ, together with the fact that $A \in \Delta_2^0$).

We want to satisfy the following requirements, which respectively ensure that $\Psi(A)$ is not r.e. and that $A \not\leq_e \Psi(A)$:

$$\begin{array}{lcl} R_{2e} & : & \Psi(A) \neq \mathcal{W}_e \\ R_{2e+1} & : & A \neq \Phi_e\Psi(A). \end{array}$$

As usual, the first type of requirement will be satisfied by the coding method (p. 511), the second by the Sacks agreement method (p. 464).

To satisfy R_{2e} we consider

$$l_1(e,s) = \max\{z : (\forall y < z)[\Psi_s(A_s)(y) = \mathcal{W}_{e,s}(y)]\}.$$

We try to code A in the e-th column, by putting $\langle e,n\rangle$ in B_1 at stage $s+1$ whenever $n \leq l_1(e,s)$. Then if $\lim_{s\to\infty} l_1(e,s) = \infty$, we have the whole e-th column in B_1 and, by definition of Ψ,

$$\langle e,n\rangle \in \Psi(A) \;\Leftrightarrow\; n \in A.$$

If $\Psi(A) = \mathcal{W}_e$, then $\Psi(A)$ is r.e. and $\lim_{s\to\infty} l_1(e,s) = \infty$ (notice that both $\Psi_s(A_s)(y)$ and $\mathcal{W}_{e,s}(y)$ have limit, the latter obviously and the former as noted above). But then A is r.e. as well, contradicting the hypothesis.

To satisfy R_{2e+1}, we let

$$\begin{array}{rcl} l_2(e,s) & = & \max\{z : (\forall y < z)[A_s(y) = \Phi_{e,s}\Psi_s(A_s)(y)]\} \\ u(e,y,s) & = & \left\{ \begin{array}{ll} \mu x.\,[A_s(y) = \Phi_{e,s}\Psi_s(A_s[x])(y)] & \text{if it exists} \\ 0 & \text{otherwise} \end{array} \right. \\ r(e,s) & = & \max\{u(e,y,s) : y \leq l_2(e,s)\}. \end{array}$$

Thus $l_2(e,s)$ monitors the length of agreement between A_s and $\Phi_{e,s}\Psi_s(A_s)$ and $r(e,s)$ shows the smallest part that one has to preserve in order to freeze $\Phi_{e,s}\Psi(A_s)$ up to the length of agreement, which is what we want to do at any stage $s+1$.

To freeze $\Phi_{e,s}\Psi_s(A_s[r(e,s)])$, recall that $\Psi_s(A_s)$ consists of $B_{0,s}$ and of the elements $\langle x,n\rangle$ of $B_{1,s}$ such that $n \in A_s$. Since A is only approximated in the limit, it might be that some $n \in A_s$ is not in A and this would affect the final value of $\Psi(A)$ (since then $\langle x,n\rangle \notin \Psi(A)$). We thus ensure that all elements $\langle x,n\rangle$ which are in $\Psi_s(A_s[r(e,s)])$, i.e. such that $n \in A_s$ and

$n \leq r(e,s)$, will stay there, by moving them from B_1 to B_0; they are thus in $\Psi(A)$, independently of A.

Then if $A = \Phi_e\Psi(A)$, we have $\lim_{s\to\infty} l_2(e,s) = \infty$ as above. It follows that $\Phi_e\Psi(A)$ is r.e., since to compute $\Phi_e\Psi(A)(y)$ it is enough to go to the least stage s such $l_2(e,s) > y$ and to note that $\Phi_e\Psi(A) = \Phi_{e,s}\Psi_s(A_s)$. But then A is r.e. as well, contradicting the hypothesis on A.

Of course, the action for requirements R_{2e+1} conflicts with the coding done for requirements R_{2e}. However, the coding does not have to succeed for all elements: it is enough for it to succeed almost always, i.e. to fail for at most finitely many elements. We can thus put together the actions for the various requirements, by using a finite injury argument.

The final construction is the following. Let $B_{0,0} = B_{1,0} = \emptyset$. At stage $s+1$, given $B_{0,s}$ and $B_{1,s}$, and for any $e \leq s$:

- put in B_1 all $\langle e,n\rangle$ such that $n \leq l_1(e,s)$
- put in B_0 all $\langle x,n\rangle \in B_{1,s}$ with $x > e$ such that $n \in A_s$ and $n \leq r(e,s)$.

The restriction $x > e$ allows R_{2e+1} to interfere only with columns of index greater than e. Thus the action for R_{2i} can be injured by the action for R_{2e+1} only if $e < i$, hence only finitely often.

Using the justification given above, it is then immediate to show by induction that each requirement is satisfied and that only finitely many elements of each column enter B. $\Box$

By V.5.11, there are minimal degrees in $\boldsymbol{\mathcal{D}}$. By XIV.2.2, there are then e-degrees minimal among the total e-degrees. By the previous result, such e-degrees cannot be minimal in $\boldsymbol{\mathcal{D}_e}$ and hence they have nonzero predecessors, which cannot be total. This provides an alternative proof of XIV.1.11.

Lagemann [1971] has strengthened the previous result by proving that if A is Δ^0_2 and not r.e., then not only is its e-degree not minimal, but it even bounds incomparable e-degrees. In particular, if there are linearly ordered initial segments of $\boldsymbol{\mathcal{D}_e}$, they cannot contain nonzero Δ^0_2 e-degrees.

Exercises XIV.2.10 (Gutteridge [1971]) a) *No total e-degree has a minimal cover.* (Hint: by the proof of XIV.2.8 it is enough to show that if C is the graph of a total function and $C <_e A \leq_T C'$, then there is Ψ such that $C <_e \Psi(A) \oplus C <_e A$. Since $A \leq_T C'$, A can be approximated recursively in C. The action for

$$\begin{array}{lcl} R^C_{2e} & : & \Psi(A) \neq \Phi_e(C) \\ R^C_{2e+1} & : & A \neq \Phi_e(\Psi(A) \oplus C) \end{array}$$

produces B which is only r.e. in C. But since C is the graph of a total function, $B \leq_e C$ because positive information from C actually decides it; to know if $\langle x,y\rangle \in C$, one enumerates C until some $\langle x,z\rangle \in C$ is found and then one compares y and z.)

b) *No total e-degree is a minimal cover*. (Hint: it is enough to show that if A is the graph of a total function and $C <_e A$, then there is Ψ such that $C <_e \Psi(A) \oplus C <_e A$. If Φ is such that $C = \Phi(A)$, the action for

$$\begin{array}{lcl} R^A_{2e} & : & \Psi(A) \neq \Phi_e(\Phi(A)) \\ R^A_{2e+1} & : & A \neq \Phi_e(\Psi(A) \oplus \Phi(A)) \end{array}$$

produces B which is only r.e. in A. But since A is the graph of a total function, $B \leq_e A$ as in part a).)

Cooper [1990a] has proved that there are e-degrees having minimal covers, so that $\boldsymbol{\mathcal{D}_e}$ *is neither dense nor homogeneous* (since in the cone above an e-degree having a minimal cover there are minimal e-degrees).

A complete characterization of the initial segments of $\boldsymbol{\mathcal{D}_e}$ is not known, but the previous results show that they are severely limited. In particular, by XIV.2.9, there is no finite initial segment.

Global properties

We turn now to global results for $\boldsymbol{\mathcal{D}_e}$. We start with the following analogue of V.7.1, which provides all the technical work.

Proposition XIV.2.11 (Slaman and Woodin [1997]) *Every countable antichain is definable from finitely many parameters in* $\boldsymbol{\mathcal{D}_e}$*, in a uniform way.*

Proof. Given an antichain $\mathcal{C} = \{\boldsymbol{c_n}\}_{n\in\omega}$ of e-degrees, we want to find parameters coding it. We adapt the proof of V.7.1 and define $\mathcal{C}$ from three parameters A, B and C (of degrees $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$), of which $C = \oplus_{n\in\omega} C_n$ for $C_n \in \boldsymbol{c_n}$. More precisely, we define $\mathcal{C}$ as

$$\boldsymbol{x} \in \mathcal{C} \;\Leftrightarrow\; \boldsymbol{x} \leq \boldsymbol{c} \wedge P(\boldsymbol{x}) \wedge \neg(\exists \boldsymbol{z} \leq \boldsymbol{c})(\boldsymbol{z} < \boldsymbol{x} \wedge P(\boldsymbol{z})),$$

where

$$P(\boldsymbol{x}) \;\Leftrightarrow\; \boldsymbol{x} \neq (\boldsymbol{x} \cup \boldsymbol{a}) \cap (\boldsymbol{x} \cup \boldsymbol{b}),$$

i.e. as the set of minimal solutions of P below $\boldsymbol{c}$. The conditions on A and B are the following:

1. $\boldsymbol{c_n}$ *is a solution of* P

   This is achieved by having, for every n, a set D_n such that

   $$D_n \leq_e C_n \oplus A, C_n \oplus B \;\text{ but }\; D_n \not\leq_e C_n.$$

   The definition of D_n in V.7.1 uses negative information about C_n and thus has to be modified. Since the sets e-reducible to C_n are of the form $\Phi_e(C_n)$, for some e, and by XIV.1.4

   $$x \in \Phi_e(C_n) \;\Leftrightarrow\; (\exists u)(\langle x, u\rangle \in \mathcal{W}_e \;\wedge\; D_u \subseteq C_n),$$

we let

$$x \in D_n \ \Leftrightarrow\ (\exists u)(\langle x, u\rangle \in A_n \ \wedge\ D_u \subseteq C_n),$$

where A_n is the n-th column of A. In particular, A will consist of elements of the form $\langle n, \langle x, u\rangle\rangle$.

Then $D_n \le_e C_n \oplus A$ by definition. $D_n \le_e C_n \oplus B$ because A_n and B_n will differ only finitely on the elements $\langle x, u\rangle$ such that $D_u \subseteq C_n$, by construction. It remains to ensure, by diagonalization, that $D_n \not\le_e C_n$, i.e. $D_n \neq \Phi_e(C_n)$ for every e.

2. ***c_n* *is a minimal solution of P***
   We will work with a fixed list $\{X_m\}_{m\in\omega}$ of the sets e-reducible to C, e.g. we can let X_m be $\Phi_m(C)$. Recall that we want, for each m,

$$D \le_e X_m \oplus A, X_m \oplus B \ \wedge\ D \not\le_e X_m \ \Rightarrow C_n \le_e X_m, \text{ for some } n.$$

   The requirements $R_{e,i,m}$ are:

$$\Phi_e(X_m \oplus A) = \Phi_i(X_m \oplus B) \not\le_e X_m \ \Rightarrow\ C_n \le_e X_m, \text{ for some } n.$$

   They will be satisfied as in the proof of the existence of minimal pairs (XIV.2.3), modified as in V.7.1.

We define A and B by special recursive coinfinite extensions θ_s and ϑ_s, which are defined only on elements of finitely many columns, and determine finite sets θ_s^+ and ϑ_s^+. Such conditions, although infinite, are actually determined by a finite amount of information; in particular, they can be recursively enumerated. In the following proof we will restrict attention to conditions of this kind.

At the end, we will let $A = \bigcup_{s\in\omega}\theta_s$ and $B = \bigcup_{s\in\omega}\vartheta_s$. We start with $\theta_0 = \vartheta_0 = \emptyset$. At stage $s+1$, let θ_s and ϑ_s be given. We do something only in the following cases:

- $s = 3\langle e, n\rangle$
  Then we diagonalize against $\Phi_e(C_n)$, by taking an x such that θ_s and ϑ_s are not yet defined on any element of the n-th column and by defining

$$\theta_{s+1}(\langle n, \langle x, u\rangle\rangle) = \vartheta_{s+1}(\langle n, \langle x, u\rangle\rangle) \simeq 0$$

  if $x \in \Phi_e(C_n)$, and

$$\theta_{s+1}(\langle n, \langle x, 0\rangle\rangle) = \vartheta_{s+1}(\langle n, \langle x, 0\rangle\rangle) \simeq 1$$

  if $x \notin \Phi_e(C_n)$.

  In the first case no $\langle x, u\rangle$ is in A_n and thus $x \notin D_n$. In the second case $\langle x, 0\rangle$ is in A_n, but $D_0 = \emptyset \subseteq C_n$ and thus $x \in D_n$. In both cases, x witnesses that $D_n \neq \Phi_e(C_n)$.

- $s = 3\langle e, i, m\rangle + 1$
  Then we try to vacuously satisfy $R_{e,i,m}$ by looking for two conditions θ and ϑ extending θ_s and ϑ_s, respectively, which e-split on some x, i.e. such that either
  $$x \in \Phi_e(X_m \oplus \theta^+) \ \wedge \ (\forall \vartheta_1 \supseteq \vartheta)[x \notin \Phi_i(X_m \oplus \vartheta_1^+)],$$
  or
  $$x \in \Phi_i(X_m \oplus \vartheta^+) \ \wedge \ (\forall \theta_1 \supseteq \theta)[x \notin \Phi_e(X_m \oplus \theta_1^+)].$$
  We also request that θ and ϑ agree on the elements $\langle n, \langle z, u\rangle\rangle$ such that $D_u \subseteq C_n$, for any $n < s$.

  If two such conditions exist, we let $\theta_{s+1} = \theta$ and $\vartheta_{s+1} = \vartheta$. Otherwise, we let $\theta_{s+1} = \theta_s$ and $\vartheta_{s+1} = \vartheta_s$.

- $s = 3\langle e, x, m\rangle + 2$
  Then, as in XIV.2.4, we ensure maximality of computations by looking for a condition θ extending θ_s and such that $x \in \Phi_e(X_m \oplus \theta^+)$.

  If such a condition exists, we let $\theta_{s+1} = \theta$ and
  $$\vartheta_{s+1}(z) \simeq \begin{cases} \vartheta_s(z) & \text{if } \vartheta_s(z)\downarrow \\ \theta(z) & \text{if } \vartheta_s(z)\uparrow \text{ and } \theta(z)\downarrow. \end{cases}$$
  Otherwise, we let $\theta_{s+1} = \theta_s$ and $\vartheta_{s+1} = \vartheta_s$.

At any stage $s+1$ the construction ensures that A_n and B_n agree on the elements $\langle z, u\rangle$ such that $D_u \subseteq C_n$, except possibly for $n \geq s = 3\langle e, i, m\rangle + 1$. Since there are only finitely many such cases, at the end A_n and B_n differ only finitely on such elements. This, together with the construction for $s = 3\langle e, n\rangle$, ensures that D_n has all the required properties.

It remains to show that $R_{e,i,m}$ is satisfied, with a proof similar to V.7.1. Suppose that at the end we have
$$\Phi_e(X_m \oplus A) = \Phi_i(X_m \oplus B) \not\leq_e X_m.$$
We have to prove that
$$C_n \leq_e X_m, \text{ for some } n.$$

Let $s = 3\langle e, i, m\rangle + 1$ and suppose there are no conditions Θ_0 and Θ_1 extending θ_s such that, for some x,
$$x \in \Phi_e(X_m \oplus \Theta_0^+) \ \wedge \ (\forall \theta \supseteq \Theta_1)[x \notin \Phi_e(X_m \oplus \theta^+)].$$
We prove, as in XIV.2.3, that
$$x \in \Phi_e(X_m \oplus A) \ \Leftrightarrow \ (\exists \theta \supseteq \theta_s)[x \in \Phi_e(X_m \oplus \theta^+)].$$

- If $x \in \Phi_e(X_m \oplus A)$, then the existence of $\theta \supseteq \theta_s$ such that $x \in \Phi_e(X_m \oplus \theta^+)$ follows from the definition of Φ_e.
- If there is $\Theta_0 \supseteq \theta_s$ such that $x \in \Phi_e(X_m \oplus \Theta_0^+)$, let $\Theta_1 = \theta_{3\langle e,x,m\rangle+2}$. By the hypothesis, there is $\theta \supseteq \Theta_1$ such that $x \in \Phi_e(X_m \oplus \theta^+)$. The construction at stage $3\langle e,x,m\rangle + 3$ thus ensures that $x \in \Phi_e(X_m \oplus A)$.

Since the conditions are recursive and can be recursively enumerated, it follows that

$$\Phi_e(X_m \oplus A) \leq_e X_m,$$

which is impossible by hypothesis.

Thus there are conditions Θ_0 and Θ_1 as above. By an interpolation argument as in V.7.1, we can suppose that Θ_0 and Θ_1 differ on just an element, say $\langle n, \langle z, u\rangle\rangle$. Moreover, since we can replace θ_s in the reasoning above by any given condition, we can also suppose that n is such that θ_s and ϑ_s are not yet defined on any element of the n-th column.

We now show that $C_n \leq_e X_m$, for that fixed n, by proving that

$$y \in C_n \;\Leftrightarrow\; (\exists \vartheta \supseteq \vartheta_s * \langle n, \langle z, 2^y\rangle\rangle)[x \in \Phi_i(X_m \oplus \vartheta^+)],$$

where by $\theta * a$ we mean the extension of θ such that $(\theta * a)^+ = \theta^+ \cup \{a\}$.

- If $y \in C_n$, then $D_{2^y} = \{y\} \subseteq C_n$. By the choice of Θ_0, $x \in \Phi_e(X_m \oplus \Theta_0^+)$. By monotonicity,

  $$x \in \Phi_e(X_m \oplus (\Theta_0 * \langle n, \langle z, 2^y\rangle\rangle)^+),$$

  too. Suppose that

  $$(\forall \vartheta \supseteq \vartheta_s * \langle n, \langle z, 2^y\rangle\rangle)[x \notin \Phi_i(X_m \oplus \vartheta^+)].$$

  Then at stage $s+1$ we could force

  $$\Phi_e(X_m \oplus A) \neq \Phi_i(X_m \oplus B),$$

  because:

  - $\Theta_0 * \langle n, \langle z, 2^y\rangle\rangle$ extends θ_s, since Θ_0 extends θ_s by hypothesis and the latter is undefined on any element of the n-th column
  - $\vartheta_s * \langle n, \langle z, 2^y\rangle\rangle$ extends ϑ_s, similarly
  - the two conditions agree on $\langle n, \langle z, 2^y\rangle\rangle$, as they have to (since, by case hypothesis, $D_{2^y} \subseteq C_n$).

- If $y \notin C_n$, then $D_{2^y} \not\subseteq C_n$. By the choice of Θ_1,
$$(\forall \theta \supseteq \Theta_1)[x \notin \Phi_e(X_m \oplus \theta^+)].$$
Suppose
$$(\exists \vartheta \supseteq \vartheta_s * \langle n, \langle z, 2^y \rangle\rangle)[x \in \Phi_i(X_m \oplus \vartheta^+)].$$
Then at stage $s+1$ we could force
$$\Phi_e(X_m \oplus A) \neq \Phi_i(X_m \oplus B),$$
because:
  - Θ_1 extends θ_s, by hypothesis
  - $\vartheta_s * \langle n, \langle z, 2^y \rangle\rangle$ extends ϑ_s, since the latter is undefined on any element of the n-th column.
  - the two conditions do not have to agree on $\langle n, \langle z, 2^y \rangle\rangle$ (since, by case hypothesis, $D_{2^y} \not\subseteq C_n$). $\square$

From the previous result we can immediately obtain the following analogues of V.7.2 and V.7.3.

Theorem XIV.2.12 Definability from parameters (Slaman and Woodin [1997]) *Every countably relation is definable from finitely many parameters in $\boldsymbol{\mathcal{D}_e}$, in a uniform way.*

Proof. As in V.7.2. $\square$

Theorem XIV.2.13 Slaman and Woodin's Theorem (Slaman and Woodin [1997]) *The first-order theory of $\boldsymbol{\mathcal{D}_e}$ has the same degree (and actually the same isomorphism type) as the theory of Second- Order Arithmetic.*

Proof. As in V.7.3. $\square$

To be able to extend to $\boldsymbol{\mathcal{D}_e}$ the results about absolute definability, homogeneity and automorphisms proved in Section V.7 for $\boldsymbol{\mathcal{D}}$ we would need some nontrivial definability result to start with, e.g. of the arithmetical e-degrees, or of the e-jump operator. Weaker versions of those results for the theory $\boldsymbol{\mathcal{D}'_e}$ of the e-degrees with e-jump can be obtained as for $\boldsymbol{\mathcal{D}'_a}$.

No general absolute definability result is known. Cooper [1990a] has proved that $\boldsymbol{\mathcal{D}_e}$ *is not homogeneous* because there are minimal covers, but no minimal e-degrees (see XIV.2.9). Cooper [199?c] has proved that *there are nontrivial automorphisms of* $\mathcal{D}_e$, but it is not known whether there are only countably many.

We conclude our study of $\boldsymbol{\mathcal{D}_e}$ by showing that it differs from all the other degree structures we have studied.

Theorem XIV.2.14 *$\boldsymbol{\mathcal{D}_e}$ is not elementarily equivalent to any of $\boldsymbol{\mathcal{D}_1}$, $\boldsymbol{\mathcal{D}_m}$, $\boldsymbol{\mathcal{D}_{tt}}$, $\boldsymbol{\mathcal{D}_{wtt}}$, $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{D}_a}$.*

Proof. $\boldsymbol{\mathcal{D}_e}$ differs from all the quoted structures because it does not have minimal degrees, while they all do, by VI.5.2, VI.2.2, VI.5.5, V.5.11 and XIII.1.6. □

Further information on $\boldsymbol{\mathcal{D}_e}$ can be obtained from Cooper [1990a], [1999].

XIV.3 Enumeration Degrees below $\mathbf{0}'_e$

In Chapter X we studied the substructure $\boldsymbol{\mathcal{R}}$ of $\boldsymbol{\mathcal{D}}$ consisting of the Σ^0_1 degrees. Since e-reducibility collapses one level of the Arithmetical Hierarchy, the following is the appropriate analogue for $\boldsymbol{\mathcal{D}_e}$.

Definition XIV.3.1 *$\boldsymbol{\mathcal{R}_e}$ is the substructure of $\boldsymbol{\mathcal{D}_e}$ consisting of the Σ^0_2 e-degrees.*

In Chapter XI we studied the substructure $\boldsymbol{\mathcal{D}(\leq \mathbf{0}')}$ of $\boldsymbol{\mathcal{D}}$ consisting of the Δ^0_2 degrees, i.e. of the degrees below $\mathbf{0}'$. It turns out that in this section we will also study the substructure $\boldsymbol{\mathcal{D}_e(\leq \mathbf{0}'_e)}$ of $\boldsymbol{\mathcal{D}_e}$ consisting of the e-degrees below $\mathbf{0}'_e$, for the following reason.

Proposition XIV.3.2 *The structures $\boldsymbol{\mathcal{R}_e}$ and $\boldsymbol{\mathcal{D}_e(\leq \mathbf{0}'_e)}$ coincide. In particular, $\boldsymbol{\mathcal{R}_e}$ is an ideal of $\boldsymbol{\mathcal{D}_e}$.*

Proof. By XIV.1.12.c, $\mathbf{0}'_e$ is the e-degree of $\overline{\mathcal{K}}$: we thus only have to prove that $A \leq_e \overline{\mathcal{K}}$ if and only if $A \in \Sigma^0_2$. If $A \leq_e \overline{\mathcal{K}}$, then

$$x \in A \;\Leftrightarrow\; (\exists u)(R(x,u) \;\wedge\; D_u \subseteq \overline{\mathcal{K}})$$

for some r.e. relation R. Since $R \in \Sigma^0_1$ and $\overline{\mathcal{K}} \in \Pi^0_1$, $A \in \Sigma^0_2$.

Conversely, if A is Σ^0_2, then it is r.e. in $\mathcal{K}$ by Post's Theorem (IV.1.14). By relativizing the Normal Form Theorem for r.e. relations (II.1.10) as in III.1.14, using the fact that $\mathcal{K}$ is r.e., one obtains an r.e. relation R such that

$$x \in A \;\Leftrightarrow\; (\exists u)(R(x,u) \;\wedge\; D_u \subseteq \overline{\mathcal{K}}),$$

i.e. $A \leq_e \overline{\mathcal{K}}$. □

Natural substructures of $\boldsymbol{\mathcal{R}_e}$ are the classes of Π^0_1 and Δ^0_2 e-degrees (recall that there is only one Σ^0_1 e-degree, namely $\mathbf{0}_e$). It turns out that we have already extensively studied the first of these substructures, because of the following simple but useful observation.

Proposition XIV.3.3 (Rogers [1967], Gutteridge [1971]) $\boldsymbol{\mathcal{R}}$ *is isomorphic to the structure of the* Π_1^0 *e-degrees. Moreover, the isomorphism preserves least and greatest elements, as well as l.u.b.'s.*

Proof. Recall from XIV.2.2 that the function $t : \boldsymbol{\mathcal{D}} \longmapsto \boldsymbol{\mathcal{D}_e}$ defined by

$$t(\boldsymbol{a}) = \text{the } e\text{-degree of } c_A, \text{ for any } A \in \boldsymbol{a}$$

provides an isomorphism of $\boldsymbol{\mathcal{D}}$ and the total e-degrees.

If A is r.e., then

$$c_A \equiv_e A \oplus \overline{A} \equiv_e \overline{A}.$$

Thus the restriction of t to $\boldsymbol{\mathcal{R}}$ is an isomorphism onto the Π_1^0 e-degrees, induced by the map $A \longmapsto \overline{A}$.

By XIV.2.2, t preserves l.u.b.'s and least element. Finally, since it sends $\mathcal{K}$ into $\overline{\mathcal{K}}$, i.e. $\mathbf{0}'_T$ into $\mathbf{0}'_e$, t preserves the greatest element. □

The previous isomorphism automatically provides a good deal of information about $\boldsymbol{\mathcal{R}_e}$. For example, any countable partial ordering is embeddable in $\boldsymbol{\mathcal{R}}$ (X.1.9) and hence in the Π_1^0 e-degrees. In particular, *the one-quantifier theory of* $\boldsymbol{\mathcal{R}_e}$ *is decidable.*

As another example, it follows from XI.2.4 that there are Δ_2^0 e-degrees that are not Π_1^0, and from XI.3.7 that actually *there are* Δ_2^0 *e-degrees incomparable with all nontrivial* Π_1^0 *e-degrees.* Cooper and Copestake [1988] have proved that, similarly, *there are* Σ_2^0 *e-degrees incomparable with all nontrivial* Δ_2^0 *e-degrees.* Thus the three structures of the Π_1^0, Δ_2^0 and Σ_2^0 e-degrees are quite distinct and we will comment on them in the following.

Exercises XIV.3.4 Total *e*-degrees below $\mathbf{0}'_e$.

a) *Every total e-degree below* $\mathbf{0}'_e$ *is* Δ_2^0. (Hint: any total e-degree contains a characteristic function c_A. Since $c_A \equiv_e A \oplus \overline{A}$, if $A \oplus \overline{A}$ is Σ_2^0, then by symmetry it is also Π_2^0.)

b) *Not every* Δ_2^0 *e-degree is total.* (Medvedev [1955a]) (Hint: see the proof of XIV.1.11.)

c) *Every* Π_1^0 *e-degree is total.* (Gutteridge [1971]) (Hint: by XIV.2.2 and XIV.3.3. More directly, by XIV.1.8, since if $A \in \Pi_1^0$, then $\overline{A} \leq_e A$.)

d) *Every 2-r.e. e-degree is total.* (Cooper [1990a]) (hint: given A and B r.e., let

$$\langle x, s\rangle \in D \ \Leftrightarrow\ x \in A_s \ \wedge\ (\exists t \geq s)(x \in B_t).$$

Then $A - B \equiv_e \overline{D}$, i.e. $A - B$ has Π_1^0 e-degree.)

Part d) is the best possible, since Cooper [1990a] has proved that *not every 3-r.e. e-degree is total.*

Exercises XIV.3.5 Low *e*-degrees. (McEvoy [1985], McEvoy and Cooper [1985]) An e-degree is **low** if its jump is $\mathbf{0}'_e$. Obviously, all low e-degrees are below $\mathbf{0}'_e$.

a) *A has low e-degree if and only if, for every $B \leq_e A$, $B \in \Delta^0_2$.* (Hint: A has low e-degree if and only if $K_A \oplus \overline{K_A} \leq_e \overline{\mathcal{K}}$, i.e. if and only if $K_A \in \Delta^0_2$.)

b) *If A is Σ^0_2 and, for every e and x,*

$$x \in \Phi_{e,s}(A_s) \textit{ for infinitely many } s \ \Rightarrow\ x \in \Phi_e(A),$$

where $\{A_s\}_{s\in\omega}$ is a recursive sequence of recursive sets such that

$$x \in A \ \Leftrightarrow\ (\exists t)(\forall s \geq t)(x \in A_s),$$

then A has low e-degree. (Hint: see X.3.3.)

c) *The isomorphism of $\boldsymbol{\mathcal{R}}$ and the Π^0_1 e-degrees preserves lowness.* (Hint: the isomorphism is the restriction to $\boldsymbol{\mathcal{R}}$ of the isomorphism of $\boldsymbol{\mathcal{D}}$ and the total e-degrees, which preserves all jumps by XIV.2.2.)

Density

Since $\boldsymbol{\mathcal{R}}$ is dense (X.6.3) and isomorphic to the Π^0_1 e-degrees, the latter are dense too. But much more is true.

Theorem XIV.3.6 Density Theorem for $\boldsymbol{\mathcal{R}_e}$ (Cooper [1984]) *Given Σ^0_2 e-degrees $\boldsymbol{c} < \boldsymbol{a}$, there is a Σ^0_2 e-degree $\boldsymbol{b}$ such that $\boldsymbol{c} < \boldsymbol{b} < \boldsymbol{a}$.*

Proof. Since the Σ^0_2 e-degrees are an ideal, in particular they are closed downwards. It is thus enough to show that if A is Σ^0_2 and $C <_e A$, then there is an enumeration operator Ψ such that $C <_e \Psi(A) \oplus C <_e A$. The obvious approach is to modify the proof of XIV.2.9 along the lines of XIV.2.10.b and satisfy the following requirements:

$$\begin{array}{lcl} R_{2e} & : & \Psi(A) \neq \Phi_e(\Phi(A)) \\ R_{2e+1} & : & A \neq \Phi_e(\Psi(A) \oplus \Phi(A)), \end{array}$$

where Φ is such that $C = \Phi(A)$.

Since the proof of XIV.2.9 uses a recursive approximation $\{A_s\}_{s\in\omega}$ of A that is correct in the limit, it seems to work only for Δ^0_2 sets. Luckily, however, the proof still goes through if the recursive approximation $\{A_s\}_{s\in\omega}$ only has the following properties:

$$x \in A \ \Leftrightarrow\ (\exists s)(\forall t \geq s)(x \in A_t)$$

and

$$(\exists_\infty s)(A_s \subseteq A).$$

Indeed, the second property ensures that infinitely often we consider approximations that produce correct values of the e-reductions to A (which, we recall, use only positive information). The first property ensures that these correct approximations actually converge to A.

More specifically, the two properties allow us to prove the following two basic facts on which the proof of XIV.2.9 rests. First, that if

$$l_1(e,s) = \max\{z : (\forall y < z)[\Psi_s(A_s)(y) = \Phi_{e,s}\Phi_s(A_s)(y)]\}$$

and $\Psi(A) = \Phi_e(\Phi(A))$, then $\lim_{s\to\infty} l_1(e,s) = \infty$. Second, that if

$$l_2(e,s) = \max\{z : (\forall y < z)[A_s(y) = \Phi_{e,s}(\Psi_s(A_s) \oplus \Phi_s(A_s))(y)]\}$$

and $A = \Phi_e(\Psi(A) \oplus \Phi(A))$, then $\lim_{s\to\infty} l_2(e,s) = \infty$. In both cases we obtain $A \leq_e \Phi(A) = C$, contradicting the hypothesis $C <_e A$.

It remains to prove that any Σ^0_2 set A admits an approximation as above. Since any Σ^0_2 set is r.e. in $\mathcal{K}$, choose e such that $A = \mathcal{W}^{\mathcal{K}}_e$, let f be a one-one enumeration of $\mathcal{K}$ and define

$$A_s = \mathcal{W}^{\mathcal{K}_s[f(s)]}_{e,s}.$$

The two properties follow easily by considering the nondeficiency stages of f (see p. I.246), i.e. the stages s such that $\mathcal{K}_s[f(s)]$ has settled on all elements $\leq f(s)$.

- If $x \in A$, i.e. $x \in \mathcal{W}^{\mathcal{K}}_e$, let s be any nondeficiency stage greater than all elements of $\mathcal{K}$ used in the computation and such that $x \in \mathcal{W}^{\mathcal{K}_s[f(s)]}_{e,s}$. Then $x \in \mathcal{W}^{\mathcal{K}_t[f(t)]}_{e,t}$, i.e. $x \in A_t$, for all $t \geq s$.

- If s is any nondeficiency stage and $x \in \mathcal{W}^{\mathcal{K}_s[f(s)]}_{e,s}$, then $x \in \mathcal{W}^{\mathcal{K}}_e$. Thus $A_s \subseteq A$. $\square$

Since Calhoun and Slaman [1996] have constructed two Π^0_2 e-degrees with no e-degree in between, *neither the Π^0_2 e-degrees nor the e-degrees below $\mathbf{0}''_e$ are dense* and the previous result is the best possible in terms of the Arithmetical Hierarchy. Lachlan and Shore [1992] have instead proved that *the n-r.e.a. e-degrees are dense*, for any $n > 1$.

Exercise XIV.3.7 *The 2-r.e.a. e-degrees are exactly the Σ^0_2 e-degrees.* (Lachlan and Shore [1992]) (Hint: a Σ^0_2 set is r.e. in $\mathcal{K}$ and above it in e-degree, because the e-degree of $\mathcal{K}$ is $\mathbf{0}_e$.)

Greatest lower bounds

Since $\boldsymbol{\mathcal{R}}$ is isomorphic to the Π^0_1 e-degrees by XIV.3.3, the existence of r.e. minimal pairs in the Turing degrees (X.6.5) implies the existence of minimal pairs in the Π^0_1 e-degrees. McEvoy and Cooper [1985] have proved that in general these are not minimal pairs in the Σ^0_2 e-degrees, but in some special cases they are: this follows from the next result, which provides an analogue of X.6.12.a.

Proposition XIV.3.8 (McEvoy and Cooper [1985]) *The isomorphism between $\boldsymbol{\mathcal{R}}$ and the Π^0_1 e-degrees preserves g.l.b.'s of low degrees.*

Proof. Let A and B be low r.e. sets and C be their g.l.b. in the T-degrees. To show that $\overline{C}$ is the g.l.b. of $\overline{A}$ and $\overline{B}$ in the e-degrees, we need to show that

$$X \leq_e \overline{A}, \overline{B} \;\Rightarrow\; X \leq_e \overline{C}.$$

Using lowness of A and B, we will define an r.e. set D such that

$$X \leq_e \overline{D} \leq_e \overline{A}, \overline{B}.$$

By the isomorphism, it then follows that $D \leq_T A, B$. Since C is the g.l.b. of A and B in the T-degrees, $D \leq_T C$. By the isomorphism again, $\overline{D} \leq_e \overline{C}$. But $X \leq_e \overline{D}$ and so $X \leq_e \overline{C}$.

It remains to prove the following lemma:

- *if A and B are low r.e. sets and $X \leq_e \overline{A}, \overline{B}$, there is an r.e. set D such that $X \leq_e \overline{D} \leq_e \overline{A}, \overline{B}$.*

Since $X \leq_e \overline{A}$, there is an r.e. relation R_A such that

$$x \in X \;\Leftrightarrow\; (\exists u)(R_A(x,u) \,\wedge\, D_u \subseteq \overline{A}).$$

The right-hand expression is r.e. in A, hence recursive in A' by completeness of the jump and recursive in $\mathcal{K}$ by lowness of A. By the Limit Lemma IV.1.17, there is a 0,1-valued recursive function f_A having always limit and such that

$$x \in X \;\Leftrightarrow\; (\exists u)(R_A(x,u) \,\wedge\, D_u \subseteq \overline{A}) \;\Leftrightarrow\; \lim_{s\to\infty} f_A(x,s) = 1.$$

Similarly, since $X \leq_e \overline{B}$ and B is low, there is an r.e. relation R_B and a 0,1-valued recursive function f_B having always limit and such that

$$x \in X \;\Leftrightarrow\; (\exists u)(R_B(x,u) \,\wedge\, D_u \subseteq \overline{B}) \;\Leftrightarrow\; \lim_{s\to\infty} f_B(x,s) = 1.$$

Let

$$\langle x,t\rangle \in \overline{D} \;\Leftrightarrow\; (\forall s \geq t)(f_A(x,s) = 1 \,\vee\, f_B(x,s) = 1).$$

D is obviously r.e. It remains to prove the following.

1. $X \leq_e \overline{D}$

   It is enough to prove

   $$x \in X \Leftrightarrow (\exists t)(\langle x, t\rangle \in \overline{D}).$$

   If $x \in X$, then e.g. $\lim_{s\to\infty} f_A(x,s) = 1$. Thus there is a stage t such that $(\forall s \geq t)(f_A(x,s) = 1)$. Hence $\langle x, t\rangle \in \overline{D}$.

   Conversely, if there is a stage t such that

   $$(\forall s \geq t)(f_A(x,s) = 1 \vee f_B(x,s) = 1),$$

   then either $f_A(x,s) = 1$ for infinitely many s and $\lim_{s\to\infty} f_A(x,s) = 1$, or $f_B(x,s) = 1$ for infinitely many s and $\lim_{s\to\infty} f_B(x,s) = 1$. In both cases, $x \in X$.

2. $\overline{D} \leq_e \overline{A}$

   Given x and t, we first ask whether $\lim_{s\to\infty} f_A(x,s) = 1$. If not, then $x \notin X$, so that $\lim_{s\to\infty} f_A(x,s) = 0$ and $\lim_{s\to\infty} f_B(x,s) = 0$. Then $\langle x, t\rangle \notin \overline{D}$.

   If instead $\lim_{s\to\infty} f_A(x,s) = 1$, we ask whether there is $s > t$ such that

   $$f_A(x,s) = 0 \wedge f_B(x,s) = 0.$$

   If not, then $\langle x, t\rangle \in \overline{D}$. Otherwise, $\langle x, t\rangle \notin \overline{D}$.

   The first question is e-reducible to $\overline{A}$, because

   $$\lim_{s\to\infty} f_A(x,s) = 1 \Leftrightarrow (\exists u)(R_A(x,u) \wedge D_u \subseteq \overline{A}).$$

   The second question is r.e. The whole procedure is thus an e-reduction of $\overline{D}$ to $\overline{A}$.

3. $\overline{D} \leq_e \overline{B}$

   As in part 2. □

Corollary XIV.3.9 (Gutteridge [1971]) *There exists a minimal pair of low e-degrees.*

Proof. By X.6.7.a, there is a minimal pair of low r.e. degrees and hence of low Π^0_1 e-degrees. □

Cooper and Sorbi [1996] have proved the **Non-Capping Theorem for $\mathcal{R}_e$**: *there is a nontrivial Σ^0_2 e-degree that is not part of a minimal pair.*

Nies and Sorbi [199?a] have proved the **Branching Theorem for $\mathcal{R}_e$**: *every Σ^0_2 e-degree is branching.*

Corollary XIV.3.10 *There are two low e-degrees without g.l.b. Hence $\mathcal{R}_e$ is not a lattice.*

Proof. The proof of XIV.3.8 actually shows that if A and B are low r.e. sets, then $\overline{A}$ and $\overline{B}$ have g.l.b. in the e-degrees if and only if they have g.l.b. in the Π^0_1 e-degrees, i.e. if and only if A and B have g.l.b. in the Turing degrees. By X.6.26, there are two low r.e. degrees without g.l.b. and hence two low Π^0_1 e-degrees without g.l.b. □

Least upper bounds

Since $\mathcal{R}$ is isomorphic to the Π^0_1 e-degrees, the Splitting Theorem for $\mathcal{R}$ (X.6.13) implies that every Π^0_1 e-degree is the l.u.b. of two incomparable low e-degrees. In particular, since $\mathbf{0}'_e$ is the e-degree of the Π^0_1 set $\overline{\mathcal{K}}$, one has:

Proposition XIV.3.11 $\mathbf{0}'_e$ *is the l.u.b. of two incomparable low e-degrees.*

Ahmad and Lachlan [1998] have proved the **Non-Splitting Theorem for $\mathcal{R}_e$**: *there is a nonzero low e-degree that is not the l.u.b. of two incomparable e-degrees.*

Cooper, Sorbi and Yi [1996] have proved the **Non-Cupping Theorem for $\mathcal{R}_e$**: *there is a nontrivial Σ^0_2 e-degree that is not part of a pair joining to $\mathbf{0}'_e$.*

Nies and Sorbi [199?] have proved that, as the *wtt*-degree of a hypersimple set is not cuppable in $\mathcal{R}_{wtt}$ (see p. 556), *the e-degree of a set hypersimple relative to $\emptyset'$ is not cuppable in $\mathcal{R}_e$.*

Lattice embeddings

On the positive side, since all lattices that are known to be embeddable in $\mathcal{R}$ (see p. 559) are actually embeddable in the low r.e. degrees, they are also embeddable in $\mathcal{R}_e$ by XIV.3.8.

On the negative side, the Non-Diamond Theorem for $\mathcal{R}$ (X.6.23) shows that the diamond cannot be embedded in the Π^0_1 e-degrees preserving the least and greatest element, but the next result shows that in $\mathcal{R}_e$ it can.

Theorem XIV.3.12 Diamond Theorem for $\mathcal{R}_e$ (Ahmad [1991]) *There are two nontrivial low e-degrees $\boldsymbol{a}$ and $\boldsymbol{b}$ such that*

$$\boldsymbol{a} \cap \boldsymbol{b} = \mathbf{0}_e \quad \textit{and} \quad \boldsymbol{a} \cup \boldsymbol{b} = \mathbf{0}'_e.$$

Proof. We build two sets A and B whose e-degrees form a low minimal pair and such that $\overline{\mathcal{K}} = A \cap B$. Then $\mathbf{0}'_e$ is the l.u.b. of the e-degrees of A and B, because

$$x \in \overline{\mathcal{K}} \Leftrightarrow x \in A \cap B \Leftrightarrow 2x \in A \oplus B \wedge 2x + 1 \in A \oplus B.$$

Since joining two low e-degrees to $\mathbf{0}'_e$ automatically ensures that they are not trivial, by XIV.3.5.b the requirements for the construction of A and B are the following:

$$\begin{array}{lcl}
N & : & x \in \mathcal{K}_{s+1} - \mathcal{K}_s \;\Rightarrow\; x \notin A_{s+1} \text{ or } x \notin B_{s+1} \\
L^A_{\langle e,x\rangle} & : & (\exists_\infty s)[x \in \Phi_{e,s}(A_s)] \;\Rightarrow\; x \in \Phi_e(A) \\
L^B_{\langle i,x\rangle} & : & (\exists_\infty s)[x \in \Phi_{i,\,s}(B_s)] \;\Rightarrow\; x \in \Phi_i(B) \\
R_{\langle e,i\rangle} & : & \Phi_e(A) = \Phi_i(B) = C \;\Rightarrow\; C \text{ r.e.}
\end{array}$$

They are ordered in a priority list, as usual, with N having the highest priority (because it ensures that $\overline{\mathcal{K}} = A \cap B$).

In this construction, contrary to the practice for r.e. Turing degrees, the coding is taken care of by a negative requirement and the restraints are produced by positive ones. Indeed, the coding requirement N tries to take elements out of A or B as soon as they appear in $\mathcal{K}$, while the lowness and minimality requirements try to keep (and, possibly, put) elements in A or B, to either decide the e-jump or preserve one side of an agreement.

The construction can be seen as a dual version of X.6.13. There we started from $A_0 = B_0 = \emptyset$ and put any element generated in $\mathcal{K}$ into A or B, according to whether the negative requirement of highest priority that would be injured tries to protect the B-side or the A-side. Here we start from $A_0 = B_0 = \omega$ and take any element generated in $\mathcal{K}$ out of A or B, according to whether the positive requirement of highest priority that would be injured tries to protect the B-side or the A-side.

More precisely, if at stage $s+1$ we see that $x \in \Phi_{e,s}(A_s)$, then $x \in \Phi_{e,s}(D)$ for some finite set $D \subseteq A_s$ and we restrain the elements of one such set D from exiting A. Similarly for Φ_i and B. Moreover, if at stage $s+1$ we see that $x \in \Phi_{e,s}(A_s) \cap \Phi_{i,s}(B_s)$, then we put x into an auxiliary set $C_{\langle e,i\rangle}$ and request that x will always remain in at least one of $\Phi_e(A)$ and $\Phi_i(B)$.

For example, if at some later stage t the extraction of some element of $\mathcal{K}$ from A causes a change of $\Phi_e(A)$ that makes $x \notin \Phi_{e,t}(A_t)$, then we restrain x from exiting $\Phi_i(B)$. Thus x becomes a witness of a disagreement of $\Phi_e(A)$ and $\Phi_i(B)$ and hence of the trivial satisfaction of $R_{\langle e,i\rangle}$. If further action makes again $x \in \Phi_e(A)$ later on, then the agreement is restored and we drop the restraint on x.

The action for N conflicts with the action for the lowness requirements, as follows. Suppose $L^A_{\langle e,x_0\rangle}$ restrains the elements of a finite set D_0 from exiting A, $L^B_{\langle i,x_1\rangle}$ restrains the elements of a finite set D_1 from exiting B and an element $z \in D_0 \cup D_1$ enters $\mathcal{K}$, so that it must be taken out of either A or B, thus injuring either $L^A_{\langle e,x_0\rangle}$ or $L^B_{\langle i,x_1\rangle}$. The obvious solution is to injure the requirement of lower priority. Since for any requirement there are only finitely many requirements of higher priority, each restraining only a finite set, eventually all lowness requirements can be permanently satisfied.

The action for N conflicts with the action for the minimality requirements, as follows. First, suppose x_0 has gone into $C_{\langle e_0,i_0\rangle}$ at a certain stage and it is now restrained from leaving $\Phi_{e_0}(A)$ because it has in the meantime left $\Phi_{i_0}(B)$, when a certain element z_0 was taken out of B. Second, suppose x_1 has gone into $C_{\langle e_1,i_1\rangle}$ at a certain stage and it is now restrained from leaving $\Phi_{i_1}(B)$ because it has in the meantime left $\Phi_{e_1}(A)$, when a certain element z_1 was taken out of A. Finally, suppose that an element z has just entered $\mathcal{K}$, so that it must be taken out of either A or B, but that taking it out of A would make x_0 leave $\Phi_{e_0}(A)$, while taking it out of B would make x_1 leave $\Phi_{i_1}(B)$. Then whatever action we take for the satisfaction of N would ruin the strategy for either $R_{\langle e_0,i_0\rangle}$ or $R_{\langle e_1,i_1\rangle}$.

The solution is the following. If we decide to take z out of A, then we also take z_0 out of A and put it back into B. This keeps z_0 out of the intersection $A \cap B$ and puts x_0 back into $\Phi_{i_0}(B)$, although it takes it out of $\Phi_{e_0}(A)$. Similarly, if we decide to take z out of B. This allows us to maintain the strategy for the minimality requirements, but it may introduce an additional difficulty: an element x may switch from $\Phi_e(A)$ to $\Phi_i(B)$ infinitely often, without ever stabilizing into one of them. That this does not happen is ensured by the lowness requirements.

More precisely, we prove that $R_{\langle e,i\rangle}$ is satisfied as follows. The set $C_{\langle e,i\rangle}$ is r.e. by construction, since

$$x \in C_{\langle e,i\rangle} \Leftrightarrow (\exists s)[x \in \Phi_{e,s}(A_s) \cap \Phi_{i,s}(B_s)].$$

It is thus enough to prove that

$$\Phi_e(A) = \Phi_i(B) = C \Rightarrow C =^* C_{\langle e,i\rangle},$$

so that C is r.e., too.

- $C \subseteq C_{\langle e,i\rangle}$
  Suppose $x \in C$, i.e. $x \in \Phi_e(A)$ and $x \in \Phi_i(B)$. By the nature of the approximations, there must be a stage s such that $x \in \Phi_{e,s}(A_s)$ and $x \in \Phi_{i,s}(B_s)$, i.e. $x \in C_{\langle e,i\rangle}$.

- $C_{\langle e,i\rangle} \subseteq^* C$
  Suppose x enters $C_{\langle e,i\rangle}$ at a stage s after which $R_{\langle e,i\rangle}$ is not going to be injured by the interaction of N with the restraining strategies relative to higher priority requirements. Then $x \in \Phi_{e,t}(A_t)$ or $x \in \Phi_{i,t}(B_t)$, for all $t \geq s$. Thus either $x \in \Phi_{e,t}(A_t)$ for infinitely many t, or $x \in \Phi_{i,t}(B_t)$ for infinitely many t. By the action taken to satisfy the lowness requirements $L^A_{\langle e,x\rangle}$ and $L^B_{\langle e,x\rangle}$, either $x \in \Phi_e(A)$ or $x \in \Phi_i(B)$, i.e. $x \in C$. $\square$

Similarly to the r.e. Turing degrees, the classes of cappable and cuppable Σ^0_2 e-degrees are not disjoint (since $\boldsymbol{a}$ and $\boldsymbol{b}$ as in the Diamond Theorem are both cappable and cuppable) and exhaustive (since every Σ^0_2 e-degree is cappable or cuppable, see Sorbi [1997]).

Corollary XIV.3.13 *For every Σ^0_2 e-degree $\boldsymbol{c}$ such that $\mathbf{0}_e < \boldsymbol{c} < \mathbf{0}'_e$ there is an incomparable low e-degree $\boldsymbol{a}$.*

Proof. As in XI.3.4. $\square$

Exercises XIV.3.14 a) *$\boldsymbol{\mathcal{R}_e}$ is not distributive.* (McEvoy and Cooper [1985]) (Hint: by a variation of X.6.27.a, the pentagon lattice can be embedded in the low r.e. degrees and hence in the low Π^0_1 e-degrees. By XIV.3.8, it can be embedded in the Σ^0_2 e-degrees, too.)

b) *$\boldsymbol{\mathcal{D}_e}$ is not distributive.* (Hint: by part a).)

Lempp and Sorbi [199?] have proved that not only the diamond, but *every finite lattice is embeddable in $\boldsymbol{\mathcal{R}_e}$, preserving both the least and the greatest element*.

Global properties

Slaman and Woodin [1997] have proved that *the first-order theory of $\boldsymbol{\mathcal{R}_e}$ is undecidable* (see Nies [1997] for a different proof), but it is not known whether it has the same degree as the theory of First-Order Arithmetic.

Turning to subclasses of sentences, it follows from XIV.3.3 that *the one-quantifier theory of $\boldsymbol{\mathcal{R}_e}$ is decidable*, and from Nies [1997] that *the three-quantifier theory is undecidable*. It is not known whether the two-quantifier theory is decidable.

We conclude our study of $\boldsymbol{\mathcal{R}_e}$ by showing that it differs from all the other r.e. degree structures we have studied.

Theorem XIV.3.15 *$\boldsymbol{\mathcal{R}_e}$ is not elementarily equivalent to any of $\boldsymbol{\mathcal{R}_1}$, $\boldsymbol{\mathcal{R}_m}$, $\boldsymbol{\mathcal{R}_{btt}}$, $\boldsymbol{\mathcal{R}_{tt}}$, $\boldsymbol{\mathcal{R}_{wtt}}$, $\boldsymbol{\mathcal{R}}$.*

Proof. $\boldsymbol{\mathcal{R}_e}$ differs from $\boldsymbol{\mathcal{R}_1}$, $\boldsymbol{\mathcal{R}_m}$, $\boldsymbol{\mathcal{R}_{btt}}$ and $\boldsymbol{\mathcal{R}_{tt}}$ because it does not have minimal degrees, while they all do by X.7.2, X.5.11, X.7.8 and X.7.16.

$\boldsymbol{\mathcal{R}_e}$ differs from $\boldsymbol{\mathcal{R}_{wtt}}$ and $\boldsymbol{\mathcal{R}}$ because of the Diamond Theorem, which fails in the latter two structures by X.6.23. $\square$

It is not known whether $\boldsymbol{\mathcal{R}_e}$ is elementarily equivalent to $\boldsymbol{\mathcal{R}_a}$. Notice that the Diamond Theorem holds in both cases, while the Density Theorem is not known to fail for $\boldsymbol{\mathcal{R}_a}$.

Since $\boldsymbol{\mathcal{R}_e}$ coincides with $\boldsymbol{\mathcal{D}_e}(\leq \mathbf{0}'_e)$, it makes sense to compare it to other structures of the same kind. The cases of 1-degrees and m-degrees are trivial, since $\boldsymbol{\mathcal{R}_1}$ and $\boldsymbol{\mathcal{R}_m}$ are ideals of $\boldsymbol{\mathcal{D}_1}$ and $\boldsymbol{\mathcal{D}_m}$, but the other ones are not.

Theorem XIV.3.16 $\boldsymbol{\mathcal{D}_e}(\leq \mathbf{0}'_e)$ *is not elementarily equivalent to* $\boldsymbol{\mathcal{D}_1}(\leq \mathbf{0}'_1)$, $\boldsymbol{\mathcal{D}_m}(\leq \mathbf{0}'_m)$, $\boldsymbol{\mathcal{D}_{btt}}(\leq \mathbf{0}'_{btt})$, $\boldsymbol{\mathcal{D}_{tt}}(\leq \mathbf{0}'_{tt})$, $\boldsymbol{\mathcal{D}_{wtt}}(\leq \mathbf{0}'_{wtt})$, $\boldsymbol{\mathcal{D}}(\leq \mathbf{0}')$ *and* $\boldsymbol{\mathcal{D}_a}(\leq \mathbf{0}'_a)$.

Proof. $\boldsymbol{\mathcal{D}_e}(\leq \mathbf{0}'_e)$ differs from all the quoted structures because it does not have minimal degrees, while they all do by X.7.2, X.5.11, X.7.8, X.7.16, Fejer and Shore [1990], XI.4.5 and XIII.1.6. □

Further information on $\boldsymbol{\mathcal{R}_e}$ can be obtained from Sorbi [1997] and the references quoted in there.

XIV.4 A Model of the Lambda Calculus ⋆

We have considered models of λ-calculus and of combinatory logic in Sections II.3 and II.5, respectively. In particular, we briefly dealt with:

1. *the set of partial functions* $\mathcal{P}$
   The main idea here is that continuous functionals on $\mathcal{P}$ can be coded by functions (their continuity modula). Thus a partial function can play the roles both of a function (the functional coded by it) and of an argument (of a functional).

2. *the set of natural numbers* ω
   The main idea here is that partial recursive functions can be coded by numbers (their indices). Thus a number can play the roles both of a function (the partial recursive function coded by it) and of an argument (of a function). In particular, application is interpreted as
   $$e \cdot x \simeq \varphi_e(x).$$

The two models are very natural, from their respective points of view (topological for the former, recursion-theoretical for the latter). But they both have disadvantages: the former because it is uncountable, while the set of λ-terms it intends to model is countable; the latter because it only models *partial* combinatory logic, since the application $\varphi_e(x)$ is not always defined, while the application of two λ-terms is always defined.

The two approaches can be combined by retaining the advantages of both without their disadvantages. The main idea is that enumeration operators can be coded as r.e. sets and r.e. sets can be coded by numbers. We thus have two equivalent choices for the model:

3. *the set of natural numbers ω*
   The main idea is that a number can play the roles both of a function (the enumeration operator coded by it) and of an argument (the r.e. set coded by it). In particular, application is interpreted as
   $$e \cdot x \;=\; \text{an index of } \Phi_e(\mathcal{W}_x).$$
4. *the set of r.e. sets $\mathcal{E}$*
   The main idea is that an r.e. set can play the roles both of a function (the enumeration operator defined by it) and of an argument (of an enumeration operator). In particular, application is interpreted as
   $$\mathcal{W}_e \cdot \mathcal{W}_x \;=\; \Phi_e(\mathcal{W}_x).$$

Of course, everything works because of the following trivial fact.

Proposition XIV.4.1 *The r.e. sets are closed under enumeration operators.*

Proof. Recall that, by Definition XIV.1.4,
$$x \in \Phi_e(\mathcal{W}_z) \;\Leftrightarrow\; (\exists u)(\langle x, u\rangle \in \mathcal{W}_e \wedge D_u \subseteq \mathcal{W}_z).$$
The right-hand-side is thus uniformly r.e. in x, e and z. $\square$

The following definition simply sets up notation.

Definition XIV.4.2 *If A is an r.e. set, then $\boldsymbol{\Phi_A}$ is the enumeration operator defined by it, i.e.*
$$x \in \Phi_A(B) \;\Leftrightarrow\; (\exists u)(\langle x, u\rangle \in A \wedge D_u \subseteq B).$$

If Ψ is an enumeration operator, then $\boldsymbol{G_\Psi}$ is a canonical r.e. set defining it, i.e.
$$\langle x, u\rangle \in G_\Psi \;\Leftrightarrow\; x \in \Psi(D_u).$$

The next proposition verifies the intuition that the enumeration operator defined by $\mathcal{W}_e$ is Φ_e and that Ψ is the enumeration operator defined by the canonical set defining it.

Proposition XIV.4.3 $\Phi_{\mathcal{W}_e} = \Phi_e$ *and* $\Phi_{G_\Psi} = \Psi$.

Proof. The first assertion is obvious, by definition of Φ_e. For the second one, we show that $\Phi_{G_\Psi}(B) = \Psi(B)$ for any B. Choose e such that $\Psi = \Phi_e$.

If $x \in \Psi(B)$, then $\langle x, u\rangle \in \mathcal{W}_e$, for some $D_u \subseteq B$. In particular, $x \in \Psi(D_u)$ (since $D_u \subseteq D_u$) and hence $\langle x, u\rangle \in G_\Psi$. Thus $x \in \Phi_{G_\Psi}(B)$.

Conversely, if $x \in \Phi_{G_\Psi}(B)$, then $\langle x, u\rangle \in G_\Psi$, for some $D_u \subseteq B$. Then $x \in \Psi(D_u)$ and hence $x \in \Psi(B)$, because $D_u \subseteq B$ and enumeration operators are monotone. $\square$

Exercise XIV.4.4 *If A is r.e., then $G_{\Phi_A} \supseteq A$, but equality does not hold in general.* (Hint: G_{Φ_A} codes $\Phi_A(D_u)$ for every u, while A might code only some relevant values. For example, if

$$\langle x, u\rangle \in A \Leftrightarrow x \in D_u \quad \text{and} \quad \langle x, u\rangle \in B \Leftrightarrow D_u = \{x\},$$

then Φ_A and Φ_B are both the identity operator, but $A \neq B$.)

The idea to define a model of λ-calculus is to interpret every closed term as an r.e. set, in such a way that β-equal terms have the same interpretation. To deal with variables we consider *environments*, i.e. functions

$$\eta : \text{Variables} \longrightarrow \mathcal{E}$$

associating to each variable x of the language of λ-calculus an r.e. set $\eta(x)$. Given an r.e. set A, we let $\eta[x := A]$ be the environment defined by

$$\eta[x := A](y) = \begin{cases} A & \text{if } y = x \\ \eta(y) & \text{otherwise.} \end{cases}$$

Definition XIV.4.5 (Plotkin [1972], Scott [1975]) *Given an environment η, we associate to any λ-term t a set $[\![t]\!]_\eta$, by induction on t, as follows:*

$$[\![t]\!]_\eta = \begin{cases} \eta(x) & \textit{if } t = x \\ \Phi_{[\![u]\!]_\eta}([\![v]\!]_\eta) & \textit{if } t = uv \\ G_{\Lambda X.\, [\![u]\!]_{\eta[x:=X]}} & \textit{if } t = \lambda x.\, u, \end{cases}$$

where $\Lambda X.\, [\![u]\!]_{\eta[x=:X]}$ is the function

$$A \in \mathcal{E} \ \longmapsto\ [\![u]\!]_{\eta[x:=A]}.$$

Notice that $[\![\lambda x.\, u]\!]_\eta$ is not interpreted directly as $\Lambda X.\, [\![u]\!]_{\eta[x=:X]}$, but indirectly as its graph, so that the interpretation of $[\![t]\!]_\eta$ is always a set. We now show that it is well-defined.

Proposition XIV.4.6 (Plotkin [1972], Scott [1975]) *For any environment η and term t, $[\![t]\!]_\eta$ is an r.e. set.*

Proof. By induction on the definition of $[\![t]\!]_\eta$:

1. $[\![x]\!]_\eta = \eta(x)$ is an r.e. set by definition of environment.
2. Since an enumeration operator sends r.e. sets to r.e. sets, to show inductively that $[\![uv]\!]_\eta = \Phi_{[\![u]\!]_\eta}([\![v]\!]_\eta)$ is an r.e. set we prove that $\Phi_X(Y)$ is an enumeration operator of X and Y. This is immediate by definition, since

$$x \in \Phi_X(Y) \ \Leftrightarrow\ (\exists u)(\langle x, u\rangle \in X \wedge D_u \subseteq Y).$$

3. Since the graph of an enumeration operator is an r.e. set, to show that $[\![\lambda x.\, u]\!]_\eta = G_{\Lambda X.\, [\![u]\!]_{\eta[x:=X]}}$ is an r.e. set we prove that $\Lambda X.\, [\![u]\!]_{\eta[x:=X]}$ is an enumeration operator. This can be done inductively on u:

   - If $u = x$, then $\Lambda X.\, [\![u]\!]_{\eta[x:=X]}$ is the identity function $\Lambda X.\, X$, which is an enumeration operator.
   - If $u = y \neq x$, then $\Lambda X.\, [\![u]\!]_{\eta[x:=X]}$ is the constant function $\Lambda X.\, \eta(y)$, which is an enumeration operator.
   - If $u = u_1 u_2$, then $\Lambda X.\, [\![u]\!]_{\eta[x:=X]}$ is $\Lambda X.\, \Phi_{[\![u_1]\!]_{\eta[x:=X]}}([\![u_2]\!]_{\eta[x:=X]})$, which is an enumeration operator by induction hypothesis, closure under composition and the fact (proved in part 2) that $\Phi_X(Y)$ is an enumeration operator.
   - If $u = \lambda y.\, u_1$, then $\Lambda X.\, [\![u]\!]_{\eta[x:=X]}$ is $\Lambda X.\, G_{\Lambda Y.\, [\![u_1]\!]_{\eta[x:=X;y:=Y]}}$, which is an enumeration operator by induction hypothesis, closure under composition and the fact that if Ψ is an enumeration operator, then so is G_Ψ, since

$$\langle z, v\rangle \in G_\Psi \;\Leftrightarrow\; z \in \Psi(D_v). \quad \Box$$

Having showed that the interpretation of terms is well-defined, it remains to check that it provides a model of λ-calculus.

Proposition XIV.4.7 (Plotkin [1972], Scott [1975]) *For any environment η, the interpretation of λ-terms defined above preserves β-equality, i.e.*

$$t_1 \overset{\beta}{=} t_2 \quad \Rightarrow \quad [\![t_1]\!]_\eta = [\![t_2]\!]_\eta .$$

Proof. The interpretation preserves applications of the β-rule, since

$$\begin{array}{rcll}
[\![(\lambda x.\, u)v]\!]_\eta & = & \Phi_{[\![\lambda x.\, u]\!]_\eta}([\![v]\!]_\eta) & \text{by definition of } [\![(\lambda x.\, u)v]\!]_\eta \\
& = & \Phi_{G_{\Lambda X.\, [\![u]\!]_{\eta[x:=X]}}}([\![v]\!]_\eta) & \text{by definition of } [\![\lambda x.\, u]\!]_\eta \\
& = & (\Lambda X.\, [\![u]\!]_{\eta[x:=X]})([\![v]\!]_\eta) & \text{because } \Phi_{G_\Psi} = \Psi \\
& = & [\![u]\!]_{\eta[x:=[\![v]\!]_\eta]} & \text{by definition of } \Lambda X.\, [\![u]\!]_{\eta[x:=X]} \\
& = & [\![u[x/v]]\!]_\eta & \text{by induction on } u.
\end{array}$$

Then, by induction on the number of reduction steps, the interpretation also preserves β-equality. $\Box$

Thus $[\![t]\!]_\eta$ does not depend on t, but only on its equivalence class w.r.t. β-equality. Moreover, if t is closed, then $[\![t]\!]_\eta$ does not depend on η either and we can simply write $[\![t]\!]$.

Exercises XIV.4.8 a) *An enumeration operator is compact and monotone.*

b) *Every enumeration operator admits a least fixed-point, which is r.e.* (Rogers [1967]) (Hint: as for II.3.15.)

c) $[\![\mathcal{Y}]\!]$ *is the least fixed-point operator on r.e. sets.* (Park [1970]) (Hint: $[\![\mathcal{Y}]\!]$ is a fixed-point operator on r.e. sets because so is $\mathcal{Y}$ for λ-terms. To show that it is actually the least fixed-point operator, proceed by induction on the well-founded partial order

$$D_u \sqsubset D_v \Leftrightarrow (\exists x)(\langle x, u\rangle \in D_v)$$

as in II.3.16.)

By XIV.4.4, the model provided above is obviously nonextensional, but it can be turned into an extensional model by using an enumeration without repetitions of the enumeration operators.

æ

Bibliography

We include here only references quoted in this volume. The enumeration is consistent with the Bibliography of Volume I, so that identity of labels in the two volumes indicates identity of reference. A complete bibliography of Recursion Theory has been edited by Hinman as Volume IV of the *Ω-Bibliography of Mathematical Logic*, Springer Verlag, 1987. On-line bibliographies can be found at the following Web addresses:

- `http://www-logic.uni-kl.de/bibl/index.html`
- `http://www.nd.edu:80/~cholak/computability/bib/bib.html`

We indicate publications by abbreviating their original names. The following is a list of full names of Russian publications and of their official translations.

1. *Algebra i Logika* (Algebra and Logic)
2. *Doklady Akademii Nauk S.S.S.R.* (Soviet Mathematics, Dokladi)
3. *Izvestiya Vysshikh Uchebnykh Zavedenij Matematika* (Soviet Mathematics, Izvestiya)
4. *Kibernetika* (Cybernetics)
5. *Matematicheskii Sbornik* (Mathematics of the U.S.S.R.)
6. *Matematicheskie Zametki* (Mathematical Notes of the Academy of Science U.S.S.R.)
7. *Problemy Kibernetiki* (Problems of Cybernetics)
8. *Problemi Peredachi Informatsii* (Problems of Information Transmission)
9. *Sibirskij Matematicheskii Zhurnal* (Siberian Mathematical Journal)
10. *Uspekhi Matematicheskikh Nauk* (Russian Mathematical Surveys).

Abraham, U., and Shore, R.A.

[1986] Initial segments of the degrees of size $\aleph_1$, *Isr. J. Math.* 53 (1986) 1–51.

Ackermann, W.

[1928] Zum Hilbertschen Aufbau der reellen Zahlen, *Math. Ann.* 9 (1928) 118–133, transl. in Van Heijenoort [1967], pp. 493–507.

[1940] Zur Widerspruchsfreiheit der Zahlentheorie, *Math. Ann.* 117 (1940) 162–194.

Addison, J.W.

[1965] The undefinability of the definable, *Not. Am. Math. Soc.* 12 (1965) 347.

Adleman, L.M., and Blum, M.

[1991] Inductive inference and unsolvability, *J. Symb. Log.* 56 (1991) 891–900.

Adleman, L.M., and Loui, M.C.

[1981] Space bounded simulation of multitape Turing machines, *Math. Syst. Th.* 14 (1981) 215–222.

Adleman, L.M., and Manders, K.

[1976] Diophantine complexity, *Proc. Symp. Found. Th. Comp.* 17 (1976) 81–88.

Ahmad, S.

[1991] Embedding the diamond in the Σ_2^0 enumeration degrees, *J. Symb. Log.* 56 (1991) 195–212.

Ahmad, S., and Lachlan, A.H.

[1998] Some special pairs of Σ_2^0 *e*-degrees, *Math. Log. Quart.* 44 (1998) 431–449.

Aho, A.V., Hopcroft, J.E., and Ullman, J.D.

[1974] *The design and analysis of computer algorithms*, Addison Wesley, 1974.

Aiello, W., Goldwasser, S., and Håstad, J.

[1986] On the power of interaction, *Proc. Symp. Found. Comp. Sci.* 27 (1986) 368–379.

Ajtai, M.

[1983] Σ_1^1 formulae on finite structures, *Ann. Pure Appl. Log.* 24 (1983) 1–48.

Allender, E.W., and Hemachandra, L.

[1992] Lower bounds for the Low Hierarchy, *J. Ass. Comp. Mach.* 39 (1992) 234–250.

Alton, D.A.

[1971] Recursively enumerable sets which are uniform for finite extensions, *J. Symb. Log.* 36 (1971) 271–287.

[1974] Iterated quotients of the lattice of recursively enumerable sets, *Proc. Lond. Math. Soc.* 28 (1974) 1–12.

[1975] A characterization of *r*-maximal sets, *Arch. Math. Log. Grund.* 17 (1975) 35–36.

[1976] Diversity of speed-ups and embeddability in computational complexity, *J. Symb. Log.* 41 (1976) 199–214.

[1980] 'Natural' programming languages and complexity measures for subrecursive programming languages: an abstract approach, *Lond. Math. Soc. Lect. Not.* 45 (1980) 248–285.

Ambos-Spies, K.

[1980] *On the structure of the recursively enumerable degrees*, Ph.D. Thesis, University of Munich, 1980.

[1984] Contiguous r.e. degrees, *Springer Lect. Not. Math.* 1104 (1984) 1–37.

[1984a] On pairs of r.e. degrees, *Trans. Am. Math. Soc.* 283 (1984) 507–531.

[1984b] An extension of the nondiamond theorem in classical and α-recursion theory, *J. Symb. Log.* 49 (1984) 586–607.

[1984c] On the structure of polynomial time degrees, *Springer Lect. Not. Comp. Sci.* 166 (1984) 198–208.

[1985b] Generators of the r.e. degrees, *Springer Lect. Not. Math.* 1141 (1985) 1–28.

[1985c] Cupping and noncapping in the r.e. weak truth table and Turing degrees, *Arch. Math. Log. Grund.* 25 (1985) 109–126.

[1985d] Antimitotic r.e. sets, *Zeit. Math. Log. Grund. Math.* 31 (1985) 461–477.

[1986] Inhomogeneities in the polynomial time degrees: the degrees of supersparse sets, *Inf. Proc. Lett.* 22 (1986) 113–117.

[1987] Polynomial time degrees of NP sets, in *Trends in Theoretical Computer Science*, Börger ed., Computer Science Press, 1987, pp. 95–142.

[1987a] Minimal pairs for polynomial time reducibilities, *Springer Lect. Not. Comp. Sci.* 270 (1987) 1–13.

[1989] Honest polynomial time reducibilities and the P = NP problem, *J. Comp. Syst. Sci.* 39 (1989) 250–281.

[1999] Polynomial-time degrees, in Griffor [1999], pp. 679–703.

[199?] Automorphism bases for $\mathcal{R}$, to appear.

Ambos-Spies, K., and Bentzien, L.

[199?] Separating NP-completeness notions under strong hypotheses, to appear.

Ambos-Spies, K., and Fejer, P.A.

[1988] Degree theoretical splitting properties of recursively enumerable sets, *J. Symb. Log.* 53 (1988) 1110–1137.

[199?] Embeddings of N_5 and the contiguous degrees, *to appear*.

Ambos-Spies, K., Fejer, P.A., Lempp, S., and Lerman, M.

[1996] Decidability of the two-quantifier theory of the recursively enumerable weak truth-table degrees and other distributive uppersemilattices, *J. Symb. Log.*, 61 (1996) 880–905.

Ambos-Spies, K., Homer, S., and Soare, R.I.

[1994] On minimal pairs and complete problems, *Theor. Comp. Sci.* 132 (1994) 229–241.

Ambos-Spies, K., Homer, S., and Yang, D.

[1990] Honest polynomial reductions and exptally sets, *Springer Lect. Not. Math.* 1432 (1990) 1–22.

Ambos-Spies, K., Jockusch, C., Shore, R.A., Soare, R.I.

[1984] An algebraic decomposition of the recursively enumerable degrees and classes equal to the promptly simple degrees, *Trans. Am. Math. Soc.* 281 (1984) 109–128.

Ambos-Spies, K., Lempp, and S. Lerman, M.

[1994] Lattice embeddings into the r.e. degrees preserving 1, *Log. Meth. Phil. Sci.* 9 (1994) 179–198.

[1994a] Lattice embeddings into the r.e. degrees preserving 0 and 1, *J. Lond. Math. Soc.* 49 (1994) 1–15.

Ambos-Spies, K., and Lerman, M.

[1986] Lattice embeddings into the recursively enumerable degrees, *J. Symb. Log.* 51 (1986) 257–272.

[1989] Lattice embeddings into the recursively enumerable degrees II, *J. Symb. Log.* 54 (1989) 735–770.

Ambos-Spies, K., and Nies, A.

[1992] Cappable recursively enumerable degrees and Post's program, *Arch. Math. Log.* 32 (1992) 51–56.

[1992a] The theory of the polynomial many-one degrees of the recursive sets is undecidable, *Springer Lect. Not. Comp. Sci.* 577 (1992) 209–218.

Ambos-Spies, K., Nies, A. and Shore, R.A.

[1992] The theory of the recursively enumerable weak truth-table degrees is undecidable, *J. Symb. Log.* 57 (1992) 864–874.

Ambos-Spies, K., and Shore, R.A.

[1993] Undecidability and 1-types in the r.e. degrees, *Ann. Pure Appl. Log.* 24 (1993) 3–37.

Ambos-Spies, K., and Soare, R.I.

[1989] The recursively enumerable degrees have infinitely many one types, *Ann. Pure Appl. Log.* 41 (1989) 1–23.

Ambos-Spies, K., and Yang, D.

[1990] Honest polynomial time degrees of elementary recursive sets, *Springer Lect. Not. Comp. Sci.* 440 (1990) 1–15.

Anderaa, S.O.

[1974] On k-tape versus $k-1$ tape real time computation, in *Complexity of computations*, Karp ed., American Mathematical Society, 1974, pp. 75–96.

Angluin, D.

[1980] Inductive inference of formal languages from positive data, *Inf. Contr.* 45 (1980) 117–135.

Angluin, D., and Smith, C.H.

[1983] Inductive Inference: theory and methods, *Comp. Surv.* 15 (1983) 237–269.

Arbib, M.A.

[1969] *Theories of abstract automata*, Prentice Hall, 1969.

Arbib, M.A., and Blum, M.

[1965] Machine dependence of degrees of difficulty, *Proc. Am. Math. Soc.* 16 (1965) 442–447.

Arslanov, M.M.

[1969] On effectively hypersimple sets, *Alg. Log.* 8 (1969) 143–153, transl. 8 (1969) 79–85.

[1979] Weakly recursively enumerable sets and limiting computability, *Probl. Meth. Cyb.* 15 (1979) 3–9.

[1982] On one hierarchy of the degrees of unsolvability, *Probl. Meth. Cyb.* 18 (1982) 10–17.

[1985a] Structural properties of the degrees below $\mathbf{0}'$, *Dokl. Acad. Nauk* 283 (1985) 270–273, transl. 32 (1985) 58–62.

[1988] On the uppersemilattice of Turing degrees below $\mathbf{0}'$, *Izv. Vyss. Uch. Zav. Mat.* 7 (1988) 27–33, transl. 7 (1988) 27–33.

[1989] Completeness in the arithmetical hierarchy and fixed points, *Alg. Log.* 28 (1989) 3–17, transl. 28 (1989) 3–16.

[1990] On the structure of degrees below $\mathbf{0}'$, *Springer Lect. Not. Math.* 1432 (1990) 23–32.

[1997] Degree structure in local degree theory, in *Complexity, Logic and Recursion Theory*, Sorbi ed., Dekker, 1997, pp. 49–74.

Arslanov, M.M., Lempp, S., and Shore, R.A.

[1996] On isolating r.e. and isolated d-r.e. degrees, *Lond. Math. Soc. Lect. Not.* 224 (1996) 61–80.

[1996a] Interpolating d-r.e. and r.e.a. degrees between r.e. degrees, *Ann. Pure Appl. Log.* 78 (1996) 29–56.

Arslanov, M.M., Nadyrov, R.F., Soloviev, V.D.

[1977] Criteria for completeness of recursively enumerable sets and some extensions of the Fixed-Point Theorem, *Izv. Vyss. Uch. Zav. Mat.* 228 (1977) 3–7, transl. 179 (1977) 1–4.

Ash, C.J.

[1986] Stability of recursive structures in arithmetical degrees, *Ann. Pure Appl. Log.* 32 (1986) 113–135.

Ausiello, G.

[1971] Abstract computational complexity and cycling computations, *J. Comp. Syst. Sci.* 5 (1971) 118–128.

Axt, P.

[1959] On a subrecursive hierarchy and primitive recursive degrees, *Trans. Am. Math. Soc.* 92 (1959) 85–105.

[1961] A note on 3-recursive functions, *Zeit. Math. Log. Grund. Log.* 7 (1961) 97–98.

[1963] Enumeration and the Grzegorczyk hierarchy, *Zeit. Math. Log. Grund. Math.* 9 (1963) 53–65.

[1965] Iteration of primitive recursion, *Zeit. Math. Log. Grund. Math.* 11 (1965) 253–255.

[1966] Iteration of relative primitive recursion, *Math. Ann.* 167 (1966) 53–55.

Baass, L.

[1972] A note on the intersection of complexity classes of functions, *Soc. Ind. Appl. Math. J. Comp.* 1 (1972) 288–289.

Baass, L., and Young, P.R.

[1973] Ordinal hierarchies and naming complexity classes, *J. Ass. Comp. Mach.* 20 (1973) 668–686.

Babai, L.

[1987] Random oracles separate PSPACE from the Polynomial Time Hierarchy, *Inf. Proc. Lett.* 26 (1987) 51–53.

[1990] E-mail and the unexpected power of interaction, *Proc. Symp. Struct. Compl.* 5 (1990) 30–44.

Babai, L., Fortnow, L., and Lund, C.

[1990] Nondeterministic exponential time has two-prover interactive protocols, *Proc. Symp. Found. Comp. Sci.* 31 (1990) 16–25.

Bachmann, H.

[1950] Die Normalfunktionen und das Problem der ausgezeichneten Folgen von Ordnungzahlen, *Viert. Natur. Gesell.* 95 (1950) 115–147.

Bacon, F.

[1620] *Novum organum sive indicia vera de interpretatione naturae*, 1620.

Baker, T., Gill, J., and Solovay, R.M.

[1975] Relativizations of the P = NP question, *S.I.A.M. J. Comp.* 4 (1975) 431–442.

Baker, T., and Selman, A.L.

[1979] A second step toward the Polynomial Time Hierarchy, *Theor. Comp. Sci.* 8 (1979) 177–187.

Balcázar, J.L.

[1985] Simplicity, relativizations and nondeterminism, *S.I.A.M. J. Comp.* 14 (1985) 148–157.

Balcázar, J.L., Book, R., and Schöning, U.

[1985] On bounded query machines, *Theor. Comp. Sci.* 40 (1985) 237–243.

[1986] The Polynomial Time Hierarchy and sparse oracles, *J. Ass. Comp. Mach.* 33 (1986) 603–617.

[1986a] Sparse sets, lowness and highness, *S.I.A.M. J. Comp.* 15 (1986) 739–747.

Balcázar, J.L., Díaz, J., and Gabarró, J.

[1988] *Structural Complexity I*, Springer, 1988.

[1990] *Structural Complexity II*, Springer, 1990.

Barashko, A.S.

[1977] Bikernels and complexity hierarchies, *Dokl. Acad. Nauk* 1 (1977) 3–5.

Barashko, A.S., and Roizen, S.J.

[1976] Kernels of partial recursive functions naming complexity classes, *Kibern.* 5 (1976) 10–15.

Bardzin, J.M.

[1965] Complexity of symmetry recognition on Turing machines, *Prob. Cyb.* 15 (1965) 245–248.

[1968] Complexity of programs to determine whether natural numbers not greater than n belong to a recursively enumerable set, *Dokl. Acad. Nauk* 182 (1968) 1249–1252, transl. 9 (1968) 1251–1254.

[1972] Prognostication of automata and functions, *Information Processing '71*, Freiman ed., North Holland 1972, 1972, pp. 81–84.

[1974] Two theorems on the limiting synthesis of functions, *Latv. Gos. Univ. Uce. Zap.* 210 (1974) 82–88.

Bardzin, J.M., and Freivalds, R.V.

[1972] On the prediction of general recursive functions, *Dokl. Acad. Nauk* 13 (1972) 1224–1228.

Bardzin, J.M., and Podnieks, K.M.

[1973] The theory of inductive inference, *Proc. Math. Found. Theor. Comp. Sci.* 1 (1973) 9–15.

Basu, S.

[1970] On the structure of subrecursive degrees, *J. Comp. Syst. Sci.* 4 (1970) 452–464.

Bennett, C., and Gill, J.

[1981] Relative to a random oracle A, $\mathrm{P}^A \neq \mathrm{NP}^A \neq \text{co-NP}^A$ with probability 1, *S.I.A.M. J. Comp.* 10 (1981) 96–113.

Bennett, J.

[1962] *On spectra*, Ph.D. Thesis, Princeton University, 1962.

Bennison, V.L.

[1980] Recursively enumerable complexity sequences and measure independence, *J. Symb. Log.* 45 (1980) 417–438.

Bennison, V.L., and Soare, R.I.

[1978] Some lowness properties and computational complexity sequences, *Theor. Comp. Sci.* 6 (1978) 233–254.

Bereczki, I.

[1952] Losung eines Markovschen Problems betreffs einer Ausdehnung des Begriffes der elementaren Funktion, *Acta Math. Acad. Sci. Hung.* 3 (1952) 197–218.

Berman, P.

[1976] On the structure of complete sets: almost everywhere complexity and infinitely often speed-up, *Proc. Symp. Found. Comp. Sci.* 17 (1976) 76–80.

[1977] *Polynomial reducibilities and complete sets*, Ph.D. Thesis, Cornell University, 1977.

[1978] Relationships between density and nondeterministic complexity of NP-complete languages, *Springer Lect. Not. Comp. Sci.* 62 (1978) 63–71.

Berman, P., and Hartmanis, J.

[1977] On isomorphism and density of NP and other complete sets, *S.I.A.M. J. Comp.* 6 (1977) 305–322.

Blum, L., and Blum, M.

[1975] Toward a mathematical theory of inductive inference, *Inf. Contr.* 28 (1975) 125–155.

Blum, M.

[1967] A machine independent theory of the complexity of recursive functions, *J. Ass. Comp. Mach.* 14 (1967) 322–336.

[1967a] On the size of machines, *Inf. Contr.* 11 (1967) 257–265.

[1971] On effective procedures for speeding up algorithms, *J. Ass. Comp. Mach.* 18 (1971) 290–305.

Blum, M., and Gill, J.
[1973] Some fruitful areas for research into complexity theory, in Rustin [1973], pp. 23–36.

Blum. M., and Marquez, I.
[1973] On complexity properties of recursively enumerable sets, *J. Symb. Log.* 38 (1973) 579–593.

Book, R.
[1972] On languages accepted in polynomial time, *S.I.A.M. J. Comp.* 1 (1972) 281–287.
[1974] Comparing complexity classes, *J. Comp. Syst. Sci.* 9 (1974) 213–229.
[1974a] Tally languages and complexity classes, *Inf. Contr.* 26 (1974) 186–193.
[1976] Translational lemmas, polynomial time and $(\log n)^j$-space, *Theor. Comp. Sci.* 1 (1976) 215–226.
[1981] Bounded query machines: on NP and PSPACE, *Theor. Comp. Sci.* 15 (1981) 27–39.
[1989] Restricted relativizations of complexity classes, *Proc. Symp. Appl. Math.* 38 (1989) 47–74.

Book, R., and Du, D.Z.
[1987] The existence and density of generalized complexity cores, *J. Ass. Comp. Mach.* 34 (1987) 718–730.

Book, R., Du, D.Z., and Russo, D.A.
[1988] On polynomial and generalized complexity cores, *Proc. Symp. Struct. Compl.* 3 (1988) 236–250.

Book, R., and Greibach, S.
[1970] Quasi real-time languages, *Math. Sys. Th.* 4 (1970) 97–111.

Book, R., Greibach, S., and Wegbreit, B.
[1970] Time and tape bounded Turing acceptors and principal AFLs, *J. Comp. Syst. Sci.* 4 (1970) 606–621.

Book, R., Long, T., and Selman, A.L.
[1984] Quantitative relativizations of complexity classes, *S.I.A.M. J. Comp.* 13 (1984) 461–487.
[1985] Qualitative relativizations of complexity classes, *J. Comp. Syst. Sci.* 30 (1985) 395–413.

Book, R., Wilson, C.B., and Xu, M.
[1982] Relativizing time, space and time-space, *S.I.A.M. J. Comp.* 11 (1982) 571–581.

Book, R., and Wrathall, C.
[1981] Bounded query machines: on NP and NP-QUERY, *Theor. Comp. Sci.* 15 (1981) 41–50.

Borodin, A.
[1972] Computational complexity and the existence of complexity gaps, *J. Ass. Comp. Mach.* 19 (1972) 158–174.

Borodin, A., Constable, R.L., and Hopcroft, J.E.
[1969] Dense and non dense families of complexity classes, *Symp. Switch. Aut. Th.* 10 (1969) 7–19.

Brandt, U.
[1988] Index sets in the arithmetical hierarchy, *Ann. Pure Appl. Log.* 37 (1988) 101–110.

Breidbart, S.
[1978] On splitting recursive sets, *J. Comp. Syst. Sci.* 17 (1978) 56–64.

Bridges, D., and Calude, C.

[1994] On recursive bounds for the exceptional values in speed-up, *Theor. Comp. Sci.* 132 (1994) 387–394.

Buchholz, W., Cichon, E.A., and Weiermann, A.

[1994] A uniform approach to fundamental sequences and hierarchies, *Math. Log. Quart.* 40 (1994) 273–286.

Buchholz, W., Feferman, S., Pohlers, W., and Sieg, W.

[1981] *Iterated inductive definitions and subsystems of analysis: recent proof-theoretical studies*, Spinger, 1981.

Buchholz, W., and Schütte, K.

[1988] *Proof Theory of impredicative systems of analysis*, Bibliopolis, 1988.

Buchholz, W., and Wainer, S.S.

[1987] Provably computable functions and the Fast Growing Hierarchy, *A.M.S. Contemp. Math.* 65 (1987) 179–198.

Buhrman, H., and Homer, S.

[1992] Superpolynomial circuits, almost sparse oracles and the Exponential Hierarchy, *Proc. Symp. Found. Soft. Techn.* 12 (1992) 116–127.

Buhrman, H., Homer, S., and Torenvliet, L.

[1991] Completeness for nondeterministic complexity classes, *Math. Syst. Th.* 24 (1991) 179–200.

Buhrman, H., and Torenvliet, L.

[1994] On the structure of complete sets, *Proc. Symp. Struct. Compl.* 9 (1994) 118–133.

Bukaraev, N.

[1981] On T-degrees of differences of recursively enumerable sets, *Dokl. Acad. Nauk* 228 (1981) 40–49, transl. 228 (1981) 40–52.

Bulitko, V.K.

[1980] Reducibility by Zhegalkin linear tables, *Sib. Math. J.* 21 (1980) 23–31, transl. 21 (1980) 332–339.

Burkhard, W.A., and Kroon, F.W.

[1971] Toward a weakly invariant complexity theory, *Symp. Switch. Aut. Th.* 12 (1971) 24–32.

Buss, J.F.

[1987] A theory of oracle machines, *Proc. Symp. Struct. Compl.* 2 (1987) 175–181.

[1988] Relativized alternation and space bounded computation, *J. Comp. Syst. Sci.* 36 (1988) 351–378.

Buss, S.R.

[1986] *Bounded Arithmetic*, Bibliopolis, 1986.

Cai, J.Y.

[1986] With probability one, a random oracle separates PSPACE from the polynomial time hierarchy, *Proc. Symp. Th. Comp.* 18 (1986) 21–29.

Cai, J.Y., Gundermann, T., Hartmanis, J., Hemachandra, L.A., Sewelson, V., Wagner K., and Wechsung, G.

[1988] The Boolean Hierarchy I: structural properties, *S.I.A.M. J. Comp.* 17 (1988) 1232–1252.

[1989] The Boolean Hierarchy II: applications, *S.I.A.M. J. Comp.* 18 (1989) 95–111.

Calhoun, W.C., and Slaman, T.A.

[1996] The Π^0_2 enumeration degrees are not dense, *J. Symb. Log.* 61 (1996) 1364–1379.

Calude, C.

[1982] Topological size of sets of partial recursive functions, *Zeit. Math. Log. Grund. Math.* 28 (1982) 455–462.

[1988] *Theories of Computational Complexity*, North Holland, 1988.

[1994] *Information and randomness*, Springer Verlag, 1994.

Calude, C., Istrate, G., and Zimand, M.

[1992] Recursive Baire classification and speedable functions, *Zeit. Math. Log. Grund. Math.* 38 (1992) 169–178.

Calude, C., and Zimand, M.

[1996] Effective category and measure in abstract complexity theory, *Theor. Comp. Sci.* 154 (1996) 307–327.

Carnap, R.

[1936] Testability and meaning, I, *Phil. Sci.* 3 (1936) 419–471.

[1937] Testability and meaning, II, *Phil. Sci.* 4 (1937) 1–40.

Carstens, H.G.

[1976] Δ_2^0 Mengen, *Arch. Math. Log. Grund.* 18 (1976) 55–65.

Case, J.

[1971] Enumeration reducibility and partial degrees, *Ann. Math. Log.* 2 (1971) 419–439.

[1974] Maximal arithmetical reducibility, *Zeit. Math. Log. Grund. Math.* 20 (1974) 261–270.

[1974a] Periodicity in generations of automata, *Math. Syst. Th.* 8 (1974) 15–32.

Case, J., and Lynes, C.

[1982] Inductive inference and language identification, *Proc. I.C.A.L.P.* (1982) 107–115.

Case, J., and Smith, C.H.

[1983] Comparison of identification criteria for machine inductive inference, *Theor. Comp. Sci.* 25 (1983) 193–220.

Cenzer, D.

[1977] Non generable r.e. sets, *Springer Lect. Not. Comp. Sci.* 56 (1977) 379–385.

Chandra, A.K., Kozen, D.C., and Stockmeyer, L.J.

[1981] Alternation, *J. Ass. Comp. Mach.* 28 (1981) 114–133.

Chandra, A.K., and Stockmeyer, L.J.

[1976] Alternation, *Proc. Symp. Found. Comp. Sci.* 17 (1976) 98–108.

Chen, K.J.

[1982] Tradeoffs in the inductive inference of nearly minimal size programs, *Inf. Contr.* 52 (1982) 68–86.

Chew, P., and Machtey, M.

[1981] A note on structure and looking-back applied to the relative complexity of computable functions, *J. Comp. Syst. Sci.* 22 (1981) 53–59.

Cholak, P.

[1994] The translation theorem, *Arch. Math. Log.* 33 (1994) 87–108.

[1995] *Automorphisms of the lattice of recursively enumerable sets*, Memoirs of the American Mathematical Society, n. 541, 1995.

Cholak, P., and Downey, R.

[1994] Recursively enumerable m- and tt-degrees. III: realizing all finite distributive lattices, *J. Lond. Math. Soc.* 50 (1994) 440–453.

[1994] Permutations and presentations, *Proc. Am. Math. Soc.* 122 (1994) 1237–1249.

Cholak, P., Downey, R., and Harrington, L.

[199?] Automorphisms of the lattice of computably enumerable sets: Σ^1_1-completeness, *to appear*.

Cholak, P., Downey, R., and Stob, M.

[1992] Automorphisms of the lattice of recursively enumerable sets: promtly simple sets, *Trans. Am. Math. Soc.* 332 (1992) 555– 570.

Cholak, P., and Harrington, L.

[199?] Jump classes and computably enumerable sets, *to appear*.

Cholak, P., and Nies, A.

[199?] Atomless r-maximal sets, *to appear*.

Chomsky, N.

[1956] Three models for the description of languages, *I.R.E. Transf. Inf. Th.* 2 (1956) 113–124.

Chomsky, N., and Miller, G.A.

[1958] Finite state languages, *Inf. Contr.* 1 (1958) 91–112.

Chong, C.T.

[1979] Generic sets and minimal α-degrees, *Trans. Am. Math. Soc.* 254 (1979) 157–169.

[1984] *Techniques of Admissible Recursion Theory*, Springer, 1984.

[1989] Recursively enumerable sets in models of Σ_2-collection, *Springer Lect. Not. Math.* 1388 (1989) 1–15.

[1989a] Hyperhypersimple sets and Δ_2-systems, *Ann. Pure Appl. Log.* 44 (1989) 25–38.

[1989b] Maximal sets and fragments of Peano Arithmetic, *Nagoya J. Math.* 115 (1989) 165–183.

[1992] Σ_n-definability without Σ_n-induction, *Trans. Am. Math. Soc.* 334 (1992) 349–363.

Chong, C.T., and Downey, R.G.

[1989] Degrees bounding minimal degrees, *Math. Proc. Cambr. Phil. Soc.* 105 (1989) 211–222.

[1990] Minimal degrees recursive in 1-generic degrees, *Ann. Pure Appl. Log.* 48 (1990) 215–225.

Chong, C.T., and Friedman, S.

[1999] Ordinal Recursion Theory, in Griffor [1999], pp. 277–300.

Chong, C.T., and Jockusch, C.G.

[1984] Minimal degrees and 1-generic sets below $\mathbf{0}'$, *Springer Lect. Not. Math.* 1104 (1984) 63–77.

Chong, C.T., and Lerman, M.

[1976] Hyperhypersimple α-r.e. sets, *Ann. Math. Log.* 9 (1976) 1–48.

Chong, C.T., and Mourad, K.J.

[1990] The degree of a Σ_n cut, *Ann. Pure Appl. Log.* 48 (1990) 227-235.

[1992] Σ_n-definable sets without Σ_n-induction, *Trans. Am. Math. Soc.* 334 (1992) 349–363.

Chong, C.T., and Yang, Y.

[1997] Σ_2-induction and infinite injury priority arguments, II: Tame Σ_2-coding and the jump operator, *Ann. Pure Appl. Log.* 84 (1997) 103–116.

[1998] Σ_2-induction and infinite injury priority arguments, I: Maximal sets and the jump operator, *J. Symb. Log.* 63 (1998) 797–814.

Chow, T.S.
[1972] On the structure of Blum measure, *Am. Fed. Inf. Proc. Soc. Conf.* 40 (1972) 503–506.

Church, A.
[1936] An unsolvable problem of elementary number theory, *Am. J. Math.* 58 (1936) 345–363, also in Davis [1965], pp. 89–107.

Cichon, E.A.
[1983] A short proof of two recently discovered independence results using recursion-theoretic methods, *Proc. Am. Math. Soc.* 87 (1983) 704–706.

Cichon, E.A., and Wainer, S.S.
[1983] The Slow-Growing and the Grzegorczyk hierarchies, *J. Symb. Log.* 48 (1983) 399–408.

Cleave, J.P.
[1963] A hierarchy of primitive recursive functions, *Zeit. Math. Log. Grund. Math.* 9 (1963) 331–346.
[1970] Some properties of recursively inseparable sets, *Zeit. Math. Log. Grund. Math.* 16 (1970) 187–200.

Cleave, J.P., and Rose, H.E.
[1967] $\mathcal{E}_n$-arithmetic, in *Sets, models and Recursion Theory*, Crossley ed., North Holland, 1967, pp. 297–308.

Cobham, A.
[1964] The intrinsic computational difficulty of functions, *Log. Meth. Phil. Sci.* 2 (1964) 24–30.

Cohen, L.J.
[1980] Inductive logic 1945–1977, in *Modern logic. A survey*, Agazzi ed., Reidel, 1980, pp. 353–375.

Cohen, P.F.
[1975] *Weak truth-table reducibility and the pointwise ordering of one-one recursive functions*, Ph.D. Thesis, University of Illinois, Urbana, 1975.

Cohen, P.F., and Jockusch, C.
[1975] A lattice property of Post's simple set, *Ill. J. Math.* 19 (1975) 450–453.

Cohen, P.J.
[1963] The independence of the continuum hypothesis, *Proc. Nat. Acad. Sci.* 50 (1963) 1143–1148.

Coles, R., Downey, R.G., and LaForte, G.
[199?] Strong reducibilites and the jump operator, *to appear*.

Constable, R.L.
[1970] On the size of programs in subrecursive formulation, *Proc. Symp. Th. Comp.* 2 (1970) 1–9.
[1971] Subrecursive programming languages II. On program size, *J. Comp. Syst. Sci.* 5 (1971) 315–334.
[1971a] Subrecursive programming languages III. The multiple recursive functions $\mathcal{R}_n$, *Proc. Symp. Comp. Aut.* 21 (1971) 393–410.
[1972] The operator gap, *J. Ass. Comp. Mach.* 19 (1972) 175–183.
[1973] Two types of hierarchy theorems for axiomatic complexity classes, in Rustin [1973], pp. 37–63.
[1973a] Type-two computational complexity, *Proc. Symp. Th. Comp.* 5 (1973) 108–121.

Constable, R.L., and Borodin, A.

[1972] Subrecursive programming languages I. Efficiency and program structure, *J. Ass. Comp. Mach.* 19 (1972) 526–568.

Constable, R.L., and Muchnik, S.S.

[1972] Subrecursive program schemata I and II, *J. Comp. Syst. Sci.* 6 (1972) 480–537.

Cook, S.A.

[1971] The complexity of theorem proving procedures, *Proc. Symp. Th. Comp.* 3 (1971) 151–158.

[1971a] Characterization of pushdown machines in terms of time-bounded computers, *J. Ass. Comp. Mach.* 18 (1971) 4–18.

[1973] A hierarchy for nondeterministic time complexity, *J. Comp. Syst. Sci.* 7 (1973) 343–353.

[1974] An observation on time-storage trade-off, *J. Comp. Syst. Sci.* 9 (1974) 308–316.

[1979] Deterministic CFL's are accepted simultaneously in polynomial time and log squared space, *Proc. Symp. Th. Comp.* 11 (1979) 338–345.

Cook, S.A., and McKenzie, P.

[1987] Problems complete for logarithmic space, *J. Alg.* 8 (1987) 385–394.

Cooper, S.B.

[1971] *Degrees of unsolvability*, Ph.D. Thesis, University of Leicester, 1971.

[1972a] Degrees of unsolvability complementary between r.e. degrees, *Ann. Math. Log.* 4 (1972) 31–73.

[1972b] Minimal upper bounds for sequences of recursively enumerable degrees, *J. Lond. Math. Soc.* 5 (1972) 445–450.

[1973] Minimal degrees and the jump operator, *J. Symb. Log.* 38 (1973) 249–271.

[1974] Minimal pairs and high recursively enumerable degrees, *J. Symb. Log.* 39 (1974) 655–660.

[1982] Partial degrees and the density problem, *J. Symb. Log.* 47 (1982) 854–859.

[1984] Partial degrees and the density problem, II: the enumeration degrees of Σ_2^0 sets are dense, *J. Symb. Log.* 49 (1984) 503–511.

[1987] Enumeration reducibility using bounded information: counting minimal covers, *Zeit. Math. Log. Grund. Math.* 33 (1987) 537–560.

[1989] The strong anticupping property for the recursively enumerable degrees, *J. Symb. Log.* 54 (1989) 527–539.

[1990] The jump is definable in the structure of the degrees of unsolvability, *Bull. Am. Math. Soc.* 23 (1990) 151–158.

[1990a] Enumeration reducibility, non-deterministic computations and relative computability of partial functions, *Springer Lect. Not. Math.* 1432 (1990) 57–110.

[1991] The density of the low$_2$ n-r.e. degrees, *Arch. Math. Log.* 31 (1991) 19–24.

[1992] A splitting theorem for the n-r.e. degrees, *Proc. Am. Math. Soc.* 115 (1992) 461–471.

[1993] Definability and global degree theory, *Springer Lect. Not. Logic* 2 (1993) 25–45.

[1994] Rigidity and definability in the noncomputable universe, *Log. Meth. Phil. Sci.* 9 (1994) 209–235.

[1996] A characterization of the jumps of minimal degrees below $\mathbf{0}'$, *Lond. Math. Soc. Lect. Not.* 224 (1996) 81–92.

[1997] Beyond Gödel's Theorem: the failure to capture information content, in *Complexity, Logic and Recursion Theory*, Sorbi ed., Dekker, 1997, pp. 93–122.

[1999] Local degree theory, in Griffor [1999], pp. 121–154.

[199?] On a conjecture of Kleene and Post, *to appear*.

[199?a] The recursively enumerable degrees are absolutely definable, *to appear*.

[199?b] The Turing universe is not rigid, *to appear*.

[199?c] Hartley Rogers' 1965 agenda, *to appear*.

Cooper, S.B., and Copestake, C.S.

[1988] Properly-Σ_2 enumeration degrees, *Zeit. Math. Log. Grund. Math.* 34 (1988) 491–522.

Cooper, S.B., and Epstein, R.L.

[1987] Complementing below recursively enumerable degrees, *Ann. Pure Appl. Log.* 34 (1987) 15–32.

Cooper, S.B., Harrington, L.A., Lachlan, A.H., Lempp, S., and Soare, R.I.

[1991] The d-r.e. degrees are not dense, *Ann. Pure Appl. Log.* 55 (1991) 125–151.

Cooper, S.B., Lempp, S., and Watson, P.

[1989] Weak density and cupping in the d-r.e. degrees, *Isr. J. Math.* 67 (1989) 137–152.

Cooper, S.B., and Seetapun, D.

[199?] On a question of Posner, *to appear*.

Cooper, S.B., and Sorbi, A.

[1996] Noncappable enumeration degrees below $\mathbf{0}'_e$, *J. Symb. Log.* 61 (1996) 1347–1363.

Cooper, S.B., Sorbi, A., and Yi, X.

[1996] Cupping and noncupping in the enumeration degrees of Σ^0_2 sets, *Ann. Pure Appl. Log.* 82 (1996) 317–342.

Cooper, S.B., and Yi, X.

[199?] Isolated *d*-r.e. degrees, *to appear*.

Copestake, C.S.

[1988] 1-genericity in the enumeration degrees, *J. Symb. Log.* 53 (1988) 878–887.

Csillag, P.

[1947] Eine Bemerkung zur Auflösung der eingeschachtelten Rekursion, *Acta Sci. Math. Szeged* 11 (1947) 169–173.

Daley, R.P.

[1976] Non complex sequences: characterizations and examples, *J. Symb. Log.* 41 (1976) 626–638.

Dantzig, G.B.

[1949] Programming of interdependent activities, II, Mathematical models, *Econometrics* 17 (1949) 200-211.

[1963] *Linear programming and extensions*, Princeton University Press, 1963.

Davis, M., ed.

[1965] *The undecidable*, Raven Press, 1965.

Dawes, A.M.

[1982] Splitting theorems for speed-up related to order of enumeration, *J. Symb. Log.* 47 (1982) 1–7.

Dedekind, R.

[1888] *Was sind und was sollen die Zahlen?*, Braunschweig, 1888.

Degtev, A.N.

[1972] Hereditary sets and tabular reducibility, *Alg. Log.* 11 (1972) 257–269, transl. 11 (1972) 145–152.

[1972a] *m*-powers of simple sets, *Alg. Log.* 11 (1972) 130–139, transl. 11 (1972) 74–80.

[1973] *tt* and *m*-degrees, *Alg. Log.* 12 (1973) 143–161, transl. 12 (1973) 78–89.

[1976] On minimal 1-degrees and *tt*-reducibility, *Sib. Math. J.* 17 (1976) 1014–1022, transl. 17 (1976) 751–757.

[1976a] Partially ordered sets of 1-degrees contained in recursively enumerable m-degrees, *Alg. Log.* 15 (1976) 249–266, transl. 15 (1976) 153–164.

[1978] On m-degrees of supersets of simple sets, *Mat. Zam.* 23 (1978) 889–893, transl. 23 (1978) 488–490.

[1978a] Decidability of the $\forall\exists$-theory of some quotient lattice of recursively enumerable sets, *Alg. Log.* 17 (1978) 134–143, transl. 17 (1978) 94–101.

[1978b] Three theorems on tt-degrees, *Alg. Log.* 17 (1978) 270–281, transl. 17 (1978) 187–194.

[1979] Some results on uppersemilattices and m-degrees, *Alg. Log.* 18 (1979) 664–679, transl. 18 (1979) 420–430.

[1979a] On truth-table type reducibilities in the theory of algorithms, *Usp. Mat. Nauk* 207 (1979) 137–168, transl. 34 (1979) 155–192.

[1981] Relationships among complete sets, *Dokl. Acad. Nauk* 228 (1981) 50–55, transl. 228 (1981) 53–61.

[1982] Comparison of linear reducibility with with other reducibilities truth-table like, *Alg. Log.* 21 (1982) 511–529, transl. 21 (1982) 339–353.

[1983] Relationships between truth-table like reducibilities, *Alg. Log.* 22 (1983) 243–259, transl. 22 (1983) 173–185.

[1983a] Relationships between truth-table like degrees, *Alg. Log.* 22 (1983) 35–52, transl. 22 (1983) 26–39.

[1985] On the uppersemilattice of disjunctive and linear degrees, *Mat. Zam.* 38 (1985) 310–316, transl. 38 (1985) 681–684.

[1989] On a question of P. Odifreddi, *Sib. Math. J.* 30 (1989) 185–187.

Dekhtiar, M.

[1969] On the impossibility of eliminating exhaustive search in computing a function relative to its graph, *Dokl. Acad. Nauk* 189 (1969) 748–751, transl. 14 (1969) 1146–1148.

[1976] On the relativization of deterministic and nondeterministic classes, *Springer Lect. Not. Comp. Sci.* 45 (1976) 255–259.

Dekker, J.C.E.

[1954] A theorem on hypersimple sets, *Proc. Am. Math. Soc.* 5 (1954) 791–796.

Dekker, J.C.E., and Myhill, J.

[1958a] Some theorems on classes of recursively enumerable sets, *Trans. Am. Math. Soc.* 89 (1958) 25–69.

[1960] Recursively equivalence types, *Univ. Cal. Publ. Math.* 3 (1960) 67–214.

Demuth, O.

[1987] A notion of semigenericity, *Comm. Math. Univ. Car.* 28 (1987) 71–84.

Demuth, O., and Kučera, A.

[1987] Remarks on1-genericity, semigenericity and related concepts, *Comm. Math. Univ. Car.* 28 (1987) 85–94.

Denisov, S.D.

[1970] On m-degrees of recursively enumerable sets, *Alg. Log.* 9 (1970) 422–427, transl. 9 (1970) 254–256.

[1974] Three theorems on elementary theories and tt-reducibility, *Alg. Log.* 13 (1974) 5–8, transl. 13 (1974) 1–2.

[1978] The structure of the uppersemilattice of recursively enumerable m-degrees and related questions, I, *Alg. Log.* 17 (1978) 643–683, transl. 17 (1978) 418–443.

Dennis Jones, E.C., and Wainer, S.S.

[1984] Subrecursive hierarchies via direct limits, *Springer Lect. Not. Math.* 1104 (1984) 117–128.

Ding, D., and Qian, L.

[1996] Isolated d-r.e. degrees are dense in r.e. degree structures, *Arch. Math. Log.* 36 (1996) 1–12.

[199?] An r.e. degree not isolating any d-r.e. degree, *to appear*.

[199?a] A splitting property for d-r.e. degrees, *to appear*.

Di Paola, R.

[1978] The α-operator gap theorem, *Arch. Math. Log. Grund.* 19 (1978) 115–129.

Dobkin, D., Lipton, R.J., and Reiss, S.

[1979] Linear programming is log-space hard for P, *Inf. Proc. Lett.* 8 (1979) 96–97.

Downey, R.G.

[1987a] Localization of a theorem of Ambos-Spies and the strong antisplitting property, *Arch. Math. Log. Grund.* 26 (1987) 127–136.

[1987b] Δ^0_2 degrees and transfer theorems, *Ill. J. Math.* 31 (1987) 419–427

[1988] Recursively enumerable m- and tt-degrees. II: the distribution of singular degrees, *Arch. Math. Log.* 27 (1988) 135–147.

[1988a] Two theorems on truth table degrees, *Proc. Am. Math. Soc.* 103 (1988) 281–287.

[1989] D-r.e. degrees and the Nondiamond Theorem, *Bull. Lond. Math. Soc.* 21 (1989) 43–50.

[1989a] Intervals and sublattices of the r.e. weak truth table degrees. I: density, *Ann. Pure Appl. Log.* 41 (1989) 1–26.

[1989b] Intervals and sublattices of the r.e. weak truth table degrees. II: nonbounding, *Ann. Pure Appl. Log.* 41 (1989) 153–172.

[1989c] Recursively enumerable m- and tt-degrees. I: the quantity of m-degrees, *J. Symb. Log.* 54 (1989) 553–567.

[1990] Notes on the $\mathbf{0}'''$ priority method with special attention to density results, *Springer Lect. Not. Math.* 1432 (1990) 111–140.

[1992] Nondiamond theorems for polynomial time reducibility, *J. Comp. Syst. Sci.* 45 (1992) 385–395.

[1993] On irreducible m-degrees, *Rend. Sem. Mat. Univ. Pol. Torino* 51 (1993) 109–112.

[199?] A note on btt-degrees, *to appear*.

Downey, R.G., Gasarch, W.I., Homer, S., and Moses, M.

[1989] On honest polynomial reductions, relativizations, and P = NP, *Proc. Symp. Struct. Compl.* 4 (1989) 196–207.

Downey, R.G., and Jockusch, C.

[1987] T-degrees, jump classes, and strong reducibilities, *Trans. Am. Math. Soc.* 301 (1987) 103–136.

Downey, R.G., and Lempp, S.

[1997] Contiguity and distributivity in the enumerable Turing degrees, *J. Symb. Log.* 62 (1997) 1215–1240.

Downey, R.G., Lempp, S., and Shore, R.A.

[1993] Highness and bounding minimal pairs, *Math. Log. Quart.* 39 (1993) 475–491.

[1996] Jumps of minimal degrees below $\mathbf{0}'$, *J. Lond. Math. Soc.* 54 (1996) 417–439.

Downey, R.G., and Nies, A.

[1997] Undecidability results for low complexity degree structures, *Proc. Symp. Struct. Compl.* 12 (1997) 128–132.

Downey, R., and Shore, R.A.

[1995] Degree-theoretic definitions of the low$_2$ recursively enumerable degrees, *J. Symb. Log.* 60 (1995) 727–756.

[1997] There is no degree invariant half-jump, *Proc. Am. Math. Soc.* 125 (1997) 3033–3037.

Downey, R.G., and Slaman, T.A.
[1989] Completely mitotic r.e. degrees, *Ann. Pure Appl. Log.* 41 (1989) 119–152.

Downey, R.G., and Stob, M.
[1986] Structural interactions of the r.e. T- and *wtt*-degrees, *Ann. Pure Appl. Log.* 31 (1986) 205–236.
[1992] Automorphisms of the lattice of recursively enumerable sets: orbits, *Adv. Math.* 92 (1992) 237–265.
[1993] Splitting theorems in Recursion Theory, *Ann. Pure Appl. Log.* 65 (1993) 1–106.

Downey, R.G., and Welch, L.V.
[1986] Splitting properties of r.e. sets and degrees, *J. Symb. Log.* 51 (1986) 88–109.

Driscoll, G.C.
[1968] Metarecursively enumerable sets and their metadegrees, *J. Symb. Log.* 33 (1968) 389–411.

Edmonds, J.
[1965] Paths, trees and flowers, *Can. J. Math.* 17 (1965) 449–467.
[1965a] Minimum partition of a matroid into independent subsets, *J. Res. Nat. Bur. Stand.* 69 (1965) 67–72.

Ehrenfeucht, A., and Mycielski, J.
[1971] Abbreviating proofs by adding new axioms, *Bull. Am. Math. Soc.* 77 (1971) 366–367.

Eilenberg, S.
[1974] *Automata, languages and machines*, Academic Press, 1974.
[1976] *Automata, languages and machines*, Volume II, Academic Press, 1976.

Epstein, R.L.
[1975] *Minimal degrees of unsolvability and the full approximation construction*, Memoirs of the American Mathematical Society, n. 163, 1975.
[1979] *Degrees of unsolvability: structure and theory*, Springer, 1979.
[1981] *Initial segments of degrees below* $\mathbf{0}'$, Memoirs of the American Mathematical Society, n. 241, 1981.

Epstein, R.L., Haas, R., and Kramer, R.
[1981] Hierarchies of sets and degrees below $\mathbf{0}'$, *Springer Lect. Not. Math.* 859 (1981) 32–48.

Ershov, Y.L.
[1964] Decidability of the elementary theory of relatively complemented distributive lattices and of the theory of filters, *Alg. Log.* 3 (1964) 17–38.
[1968] A hierarchy of sets, *Alg. Log.* 7 (1968) 47–74, transl. 7 (1968) 25–43.
[1969] Hyperhypersimple m-degrees, *Alg. Log.* 8 (1969) 523–552, transl. 8 (1969) 298–315.
[1970] A hierarchy of sets III, *Alg. Log.* 9 (1970) 34–51, transl. 9 (1970) 20–31.
[1970a] On inseparable pairs, *Alg. Log.* 9 (1970) 661–666, transl. 9 (1970) 396–399.
[1970b] On index sets, *Sib. Math. J.* 11 (1970) 326–342, transl. 11 (1970) 246–258.
[1971] Positive equivalences, *Alg. Log.* 10 (1971) 620–650, transl. 10 (1971) 378–394.
[1980] *Decidability problems and constructive models*, Nauka, 1980.

Ershov, Y.L., and Lavrov, I.A.
[1973] The uppersemilattice $L(\gamma)$, *Alg. Log.* 12 (1973) 167–189, transl. 12 (1973) 93–106.

Even, S., Selman, A., and Yacobi, Y.

[1985] Hard-core theorems for complexity classes, *J. Ass. Comp. Mach.* 32 (1985) 205–217.

Fabian, R.J., and Kent, C.F.

[1969] Recursive functions defined by ordinal recursions, *Proc. Am. Math. Soc.* 23 (1969) 206–210.

Facchini, E., and Maggiolo-Schettini, A.

[1982] Comparing hierarchies of primitive recursive sequence functions, *Zeit. Math. Log. Grund. Math.* 28 (1982) 431–445.

Fagin, R.

[1974] Generalized first-order spectra and polynomial time recognizable sets, in *Complexity of computation*, Karp ed., American Mathematical Society, 1974, pp. 43–73.

[1993] Finite model theory: a personal perspective, *Theor. Comp. Sci.* 116 (1993) 3–31.

Fairtlough, M.V.H., and Wainer, S.S.

[1992] Ordinal complexity of recursive definitions, *Inf. Comp.* 99 (1992) 123–153.

[1998] Hierarchies of provably recursive functions, in *Handbook of Proof Theory*, Buss ed., North Holland, 1998, pp. 149–207.

Feferman, S.

[1962] Classifications of recursive functions by means of hierarchies, *Trans. Am. Math. Soc.* 104 (1962) 101–122.

[1964] Systems of predicative analysis, *J. Symb. Log.* 29 (1964) 1–30.

[1965] Some applications of the notion of forcing and generic sets, *Fund. Math.* 56 (1965) 325–345.

Feferman, S., Scott, D., and Tennebaum, S.

[1959] Models of arithmetic through function rings, *Not. Am. Math. Soc.* 173 (1959) 159–160.

Fejer, P.A.

[1982] Branching degrees above low degrees, *Trans. Am. Math. Soc.* 273 (1982) 157–180.

[1983] The density of the nonbranching degrees, *Ann. Math. Log.* 24 (1983) 113–130.

[1989] Embedding lattices with top preserved below non-$\mathbf{GL}_2$ degrees, *Zeit. Math. Log. Grund. Math.* 35 (1989) 3–14.

Fejer, P.A., and Shore, R.A.

[1985] Embeddings and extensions of embeddings in the r.e. *tt*- and *wtt*-degrees, *Springer Lect. Not. Math.* 1141 (1985) 121–140.

[1988] Infima of recursively enumerable truth table degrees, *Notre Dame J. Form. Log.* 29 (1988) 420–437.

[1990] A direct construction of a minimal recursively enumerable truth-table degree, *Springer Lect. Not. Math.* 1432 (1990) 187–204.

[199?] Every *tt*-incomplete r.e. *tt*-degree is branching, *to appear*.

Fejer, P.A., and Soare, R.I.

[1981] The plus-cupping theorem for the recursively enumerable degrees, *Springer Lect. Not. Math.* 859 (1981) 49–62.

Feldman, J.

[1972] Some decidability results on grammatical inference and complexity, *Inf. Contr.* 20 (1972) 244–262.

Fenner, S., Fortnow, L., and Kurtz, S.A.

[1992] An oracle relative to which the isomporphism conjecture holds, *Proc. Symp. Found. Comp. Sci.* 33 (1992) 29–37.

Fenner, S., Kurtz, S.A., and Royer, J.

[1989] Every polynomial time 1-degree collapses if and only if P = PSPACE, *Proc. Symp. Found. Comp. Sci.* 30 (1989) 624–629.

Fenner, S., and Schaefer, M.

[1999] Bounded immunity and *btt*-reductions, *Math. Log. Quart.* 45 (1999) 3–21.

Fischer, P.

[1986] Pairs without infimum in the r.e. *wtt*-degrees, *J. Symb. Log.* 51 (1986) 117–129.

Fischer, P.C.

[1963] A note on bounded truth-table reducibility, *Proc. Am. Math. Soc.* 14 (1963) 875–877.

[1967] Turing machines with a schedule to keep, *Inf. Contr.* 11 (1967) 138–146.

Flajolet, P., and Steyaert, J.M.

[1974] On sets having only hard subsets, *Springer Lect. Not. Comp. Sci.* 14 (1974) 446–457.

Fortnow, L., Jain, S., Gasarch, W.I., Kinber, E.B., Kummer, M., Kurtz, S.A., Pleszkoch, M., Slaman, T.A., Stephan, F., and Solovay, R.M.

[1994] Extremes in the degrees of inferability, *Ann. Pure Appl. Log.* 66 (1994) 231–276.

Fortnow, L., and Sipser, M.

[1988] Are there interactive protocols for co-NP languages?, *Inf. Proc. Lett.* 28 (1988) 249–251.

Fortune, S.

[1979] A note on sparse complete sets, *S.I.A.M. J. Comp.* 8 (1979) 431–433.

Freivalds, R.V.

[1975] Minimal Gödel numbers and their identification in the limit, *Springer Lect. Not. Comp. Sci.* 32 (1975) 219–225.

[1990] Inductive inference of minimal programs, in *Proceedings of the Third Workshop on Computational Learning Theory*, Fulk and Case eds., Morgan Kaufmann, 1990, pp. 3–20.

Freivalds, R.V., Kinber, E.B., and Wiehagen, R.

[1982] Inductive inference and computable one-one numberings, *Zeit. Math. Log. Grund. Math.* 28 (1982) 463–479.

[1984] Connections between identifying functionals, standardizing operations, and computable numberings, *Zeit. Math. Log. Grund. Math.* 30 (1984) 145–164.

Freivalds, R.V., and Wiehagen, R.

[1979] Inductive inference with additional information, *J. Inf. Proc. Cyb.* 15 (1979) 179–185.

Friedberg, R.M.

[1957] The fine structure of degrees of unsolvability of recursively enumerable sets, *Talks Cornell Summ. Inst. Symb. Log.*, Cornell, 1957, pp. 404–406.

[1957a] Two recursively enumerable sets of incomparable degrees of unsolvability, *Proc. Nat. Acad. Sci.* 43 (1957) 236–238.

[1957b] A criterion for completeness of degrees of unsolvability, *J. Symb. Log.* 22 (1957) 159–160.

[1958] Three theorems on recursive enumerations, *J. Symb. Log.* 23 (1958) 309–316.

Friedberg, R.M., and Rogers, H.
[1959] Reducibility and completeness for sets of integers, *Zeit. Math. Log. Grund. Math.* 5 (1959) 117–125.

Friedman, H.
[1977] Classically and intuitionistically provably recursive functions, *Springer Lect. Not. Math.* 669 (1977) 21–27.

Friedman, S., and Sacks, G.E.
[1977] Inadmissible Recursion Theory, *Bull. Am. Math. Soc.* 83 (1977) 255–256.

Fukuyama, M.
[1968] A remark on minimal degrees, *Sci. Rep. Tokyo Daig.* 9 (1968) 255–256.

Fulk, M.A.
[1988] Saving the phenomena: requirements that inductive inference machines not contradict known data, *Inf. Comp.* 79 (1988) 193–209.
[1990] Prudence and other conditions on formal language learning, *Inf. Contr.* 85 (1990) 1–11.
[1990a] A note on almost everywhere h-complex functions, *J. Comp. Syst. Sci.* 40 (1990) 444–449.

Fürer, M.
[1982] The tight deterministic time hierarchy, *Proc. Symp. Th. Comp.* 14 (1982) 8–16.

Furst, M., Saxe, J., and Sipser, M.
[1984] Parity, circuits, and the Polynomial Time Hierarchy, *Math. Syst. Th.* 17 (1984) 13–27.

Gandy, R.O.
[1984] Some relations between classes of low computational complexity, *Bull. Lond. Math. Soc.* 16 (1984) 127–134.

Ganesan, K., and Homer, S.
[1989] Complete problems and strong polynomial reducibilities, *Springer Lect. Not. Comp. Sci.* 349 (1989) 240–250.

Garey, M.R., and Johnson, D.S.
[1979] *Computers and intractability. A guide to the theory of* NP*-completeness*, Freeman and Company, 1979.

Gasarch, W.I., and Homer, S.
[1983] Relativizations comparing NP and Exponential Time, *Inf. Contr.* 58 (1983) 88–100.

Gasarch, W.I., and Pleszkoch, M.
[1989] Learning via queries to an oracle, *Proc. Conf. Comp. Learn. Th.* 2 (1989) 214–229.

Gasarch, W.I., and Smith, C.H.
[1995] Recursion theoretical models of learning: some results and intuitions, *Ann. Math. Art. Int.* 15 (1995) 151–166.
[1997] A survey of inductive inference with an emphasis on queries, in *Complexity, logic and Recursion Theory*, Sorbi ed., Dekker, 1997, pp. 225–260.

Gentzen, G.
[1936] Die Widerspruchfreiheit der reinen Zahlentheorie, *Math. Ann.* 122 (1936) 493–565, transl. in [1969], pp. 132–213.
[1969] *The Collected Papers of Gerhard Gentzen*, Szabo ed., North Holland, 1969.

Geske, J., and Grollmann, J.

[1986] Relativizations of unambiguous and random polynomial time classes, *S.I.A.M. J. Comp.* 15 (1986) 511–519.

Gill, J.

[1977] Computational complexity of probabilistic Turing machines, *S.I.A.M. J. Comp.* 6 (1977) 675–695.

Gill, J., and Blum, M.

[1974] On almost everywhere complex recursive functions, *J. Ass. Comp. Mach.* 21 (1974) 425–435.

Gill, J., and Morris, P.

[1974] On subcreative sets and s-reducibility, *J. Symb. Log.* 39 (1974) 669–677.

Gill, J., and Simon, I.

[1976] Ink, dirty tape Turing machines, and quasicomplexity measures, *Int. Coll. Aut. Lang. Progr.* 3 (1976) 285–306.

Ginsburg, S.

[1975] *Algebraic and automata-theoretic properties of formal languages*, North Holland, 1975.

Girard, J.Y.

[1981] Π^1_2-logic, I, *Ann. Math. Log.* 21 (1981) 75–219.

[1987] *Proof Theory and Logical Complexity*, Bibliopolis, 1987.

Glinert, E.P.

[1971] On restricted Turing computability, *Math. Syst. Th.* 5 (1971) 331–343.

Gödel, K.

[1931] Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, *Monash. Math. Phys.* 38 (1931) 173–198, transl. in [1986], pp. 145–195, also in Davis [1965], pp. 5–38.

[1933a] Zur intuitionistischen Arithmetik und Zahlentheorie, *Ergeb. Math. Koll.* 4 (1933) 34–38, transl. in [1986], pp. 287–295.

[1936] Über die Lange der Beweise, *Ergeb. Math. Koll.* 7 (1936) 23–24, transl. in [1986], pp. 397–399, also in Davis [1965], pp. 82–83.

[1958] Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes, *Dial.* 12 (1958) 280–287, transl. in [1990], pp. 241–251.

[1986] *Collected works*, Volume I, Oxford University Press, 1986.

[1990] *Collected works*, Volume II, Oxford University Press, 1990.

Goetze, B.

[1976] The structure of the lattice of recursive sets, *Zeit. Math. Log. Grund. Math.* 22 (1976) 187–191.

Goetze, B., and Nehrlich, W.

[1978] Loop programs and classes of primitive recursive functions, *Springer Lect. Not. Comp. Sci.* 64 (1978) 232–238.

[1980] The structure of loops programs and the subrecursive hierarchy, *Zeit. Math. Log. Grund. Math.* 26 (1980) 255–278.

[1981] The number of loops necessary and sufficient for computing simple functions, *Elek. Inf. Kibern.* 17 (1981) 363–376.

Gold, E.M.

[1967] Language identification in the limit, *Inf. Contr.* 10 (1967) 447–474.

Goodman, N.

[1954] *Facts, fiction and forecast*, 1954.

Goodstein, R.L.
[1944] On the restricted ordinal theorem, *J. Symb. Log.* 9 (1944) 33–41.

Grädel, E.
[1991] Capturing complexity classes by fragments of second-order logic, *Proc. Symp. Struct. Compl.* 6 (1991) 341–352.

Grant, P.W.
[1980] Some more independence results in Complexity Theory, *Theor. Comp. Sci.* 12 (1980) 119–126.

Grassin, J.
[1974] Index sets in Ershov's hierarchy, *J. Symb. Log.* 39 (1974) 97–104.

Greenlaw, R., Hoover, H.J., and Ruzzo, W.L.
[1995] *Limits to parallel computation:* P*-completeness theory*, Oxford University Press, 1995.

Griffor, E.
[1999] *Handbook of Recursion Theory*, North Holland, 1999.

Grollmann, J, and Selman, A.L.
[1984] Complexity measures for public-key cryptosystems, *Proc. Symp. Found. Comp. Sci.* 25 (1984) 495–503.

Groszek, M., and Hummel, T.
[199?] Σ_2^0-constructions and $I\Sigma_1$, *to appear*.

Groszek, M., and Mytilinaios, M.
[1990] Σ_2-induction and the construction of a high degree, *Springer Lect. Not. Math.* 1432 (1990) 205–221.

Groszek, M., Mytilinaios, M., and Slaman, T.A.
[1996] The Sacks density theorem and Σ_2-bounding, *J. Symb. Log.* 61 (1996) 450–467.

Groszek, M., and Slaman, T.A.
[199?] On Turing reducibility, *to appear*.
[199?a] Foundations of the Priority Method, I: finite injury, *to appear*.

Grzegorczyk, A.
[1953] Some classes of recursive functions, *Rozpr. Mat.* 4 (1953) 1–45.

Gurevich, Y.
[1983] Algebra of feasible functions, *Proc. Symp. Found. Comp. Sci.* 24 (1983) 210–214.

Gutteridge, L.
[1971] *Some results on enumeration reducibility*, Ph.D. Thesis, Simon Frazer University, 1971.

Hájek, P.
[1979] Arithmetical hierarchy and complexity of computations, *Theor. Comp. Sci.* 8 (1979) 227–237.

Hájek, P., and Kučera, A.
[1989] On recursion theory in $I\Sigma_1$, *J. Symb. Log.* 54 (1989) 576–589.

Hájek, P., and Pudlák, P.
[1993] *Metamathematics of First-Order Arithmetic*, Springer, 1993.

Halpern, J.Y., Loui, M.C., Meyer, A.R., and Weise, D.
[1986] On time versus space III, *Math. Sys. Th.* 19 (1986) 13–28.

Hammond, T.

[1990] *The lattice of sets recursively enumerable in an oracle*, Ph.D. Thesis, University of California, Berkeley, 1990.

Harding, G.H.

[1974] *Forcing in recursion theory*, Ph.D. Thesis, University of Swansea, 1974.

Hardy, G.H.

[1904] A theorem concerning the infinite cardinal numbers, *Quart. J. Math.* 35 (1904) 87–94.

Harrington, L.A.

[1976] Arithmetically incomparable arithmetic singletons, *Mimeographed Notes*, 1976.

[1976a] McLaughlin's conjecture, *Mimeographed Notes*, 1976.

[1978] Plus cupping in the recursively enumerable degrees, *Mimeographed Notes*, 1978.

[1982] A gentle approach to priority arguments, *Mimeographed Notes*, 1982.

[1983] The undecidability of the lattice of recursively enumerable sets, *Mimeographed Notes*, 1983.

Harrington, L.A., and Haught, C.A.

[199?] Limitations on initial segment embeddings in the r.e. *tt*-degrees, *to appear*.

Harrington, L.A., and Kechris, A.S.

[1975] A basis result for Σ^0_3 sets of reals with an application to minimal covers, *Proc. Am. Math. Soc.* 53 (1975) 445–448.

Harrington, L.A., Lachlan, A.H., Maass, W., and Soare, R.I.

[199?] New methods for automorphisms of the recursively enumerable sets and applications to the low$_2$ sets, *to appear*.

Harrington, L.A., and Nies, A.

[1998] Coding in the partial order of enumerable sets, *Adv. Math.* 133 (1998) 133–162.

Harrington, L.A., and Shelah, S.

[1982] The undecidability of the recursively enumerable degrees, *Bull. Am. Math. Soc.* 6 (1982) 79–80.

Harrington, L.A., and Shore, R.A.

[1981] Definable degrees and automorphisms of $\mathcal{D}$, *Bull. Am. Math. Soc.* 4 (1981) 97–100.

Harrington, L.A., and Soare, R.I.

[1991] Post's program and incomplete recursively enumerable sets, *Proc. Nat. Acad. Sci.* 88 (1991) 10242–10246.

[1996] Dynamic properties of computably enumerable sets, *Lond. Math. Soc. Lect. Not.* 224 (1996) 105–121.

[1996a] Definability, automorphisms, and dynamic properties of computable enumerable sets, *Bull. Symb. Log.* 2 (1996) 199–213.

[199?] Definable properties of the computably enumerable sets, *to appear*.

[199?a] Martin's invariance conjecture and low sets, *to appear*.

[199?b] The Δ^0_3-automorphism method and noninvariant classes of degrees, *to appear*.

[199?c] Codable sets and orbits of computably enumerable sets, *to appear*.

Harrison, J.

[1968] Recursive pseudo well-orderings, *Trans. Am. Math. Soc.* 131 (1968) 526–543.

Harrow, K.

[1975] Small Grzegorczyk classes and limited minimum, *Zeit. Math. Log. Grund. Math.* 21 (1975) 417–426.

[1979] Equivalence of some hierarchies of primitive recursive functions, *Zeit. Math. Log. Grund. Math.* 25 (1979) 411–418.

Hartmanis, J.

[1968] Computational complexity of one-tape Turing machine computations, *J. Ass. Comp. Mach.* 15 (1968) 325–339.

[1973] On the problem of finding natural complexity measures, *Math. Found. Comp. Sci.* 2 (1973) 95–103.

[1978] On log-tape isomorphism of complete sets, *Theor. Comp. Sci.* 7 (1978) 273–286.

[1978a] *Feasible computations and provable complexity properties*, Philadelphia, 1978.

[1979] Relations between diagonalization proof systems, and complexity gaps, *Theor. Comp. Sci.* 8 (1979) 239–253.

[1983] On sparse sets in NP $-$ P, *Inf. Proc. Lett.* 2 (1983) 55–60.

[1985] Solvable problems with conflicting relativizations, *Bull. Europ. Ass. Theor. Comp. Sci.* 27 (1985) 40–49.

Hartmanis, J., and Berman, P.

[1978] On polynomial time isomorphism of some new complete sets, *J. Comp. Syst. Sci.* 16 (1978) 418–422.

Hartmanis, J., Chang, R., Chari, S., Ranjan, D., and Rohatgi, P.

[1992] Relativization: a revisionistic perspective, *Bull. Europ. Ass. Theor. Comp. Sci.* 47 (1992) 144–153.

Hartmanis, J., Chang, R., Kadin, J., and Mitchell, S.

[1988] Some observations about relativization of space bounded computations, *Bull. Europ. Ass. Theor. Comp. Sci.* 35 (1988) 82–92.

Hartmanis, J., Chang, R., Ranjan, D., and Rohatgi, P.

[1990] On IP = PSPACE and theorems with narrow proofs, *Bull. Europ. Ass. Theor. Comp. Sci.* 41 (1990) 166–174.

Hartmanis, J., and Hemachandra, L.A.

[1987] One-way functions, robustness, and the nonisomorphism of NP-complete sets, *Proc. Symp. Struct. Compl.* 2 (1987) 160–173.

Hartmanis, J., and Hopcroft, J.E.

[1976] Independence results in Computer Science, *S.I.G.A.C.T. News* 8 (1976) 13–24.

Hartmanis, J., and Hunt, H.B.

[1974] The LBA problem and its importance in the theory of computing, *S.I.A.M.-A.M.S. Proc.* 7 (1974) 1–26.

Hartmanis, J., and Immerman,

[1985] On complete problems for NP $\cap$ co-NP, *Springer Lect. Not. Comp. Sci.* 195 (1985) 250–259.

Hartmanis, J., Immerman, N., and Sewelson, V.D.

[1983] Sparse sets in NP $-$ P: EXPTIME versus NEXPTIME, *Proc. Symp. Th. Comp.* 15 (1983) 382–391.

Hartmanis, J., and Lewis, F.D.

[1971] The use of lists in the study of undecidable problems in automata theory, *J. Comp. Syst. Sci.* 5 (1971) 54–66.

Hartmanis, J., Lewis, F.D., and Stearns, R.E.

[1965] Hierarchies of memory limited computations, *Proc. Symp. Switch. Circ. Th. Log. Des.* 6 (1965) 179–190.

Hartmanis, J., and Mahaney, S.R.

[1980] An essay about research on sparse NP-complete sets, *Springer Lect. Not. Comp. Sci.* 88 (1980) 40–57.

Hartmanis, J., and Stearns, R.E.

[1965] On the computational complexity of algorithms, *Trans. Am. Math. Soc.* 117 (1965) 285–306.

Hartmanis, J., and Yesha, Y.

[1984] Computation times of NP sets of different densities, *Theor. Comp. Sci.* 34 (1984) 17–32.

Håstad, J.

[1986] Almost optimal lower bounds for small depth circuits, *Proc. Symp. Th. Comp.* 18 (1986) 6–20.

[1987] *Computational limitations for small-depth circuits*, M.I.T. Press, 1987.

Haught, C.A.

[1986] The degrees below a 1-generic degree and less than $\mathbf{0}'$, *J. Symb. Log.* 51 (1986) 770–777.

[1987] Lattice embeddings in the recursively enumerable *tt*-degrees, *Trans. Am. Math. Soc.* 301 (1987) 515–535.

Haught, C.A., and Shore, R.A.

[1990] Undecidability and initial segments of the (r.e.) *tt*-degrees, *J. Symb. Log.* 55 (1990) 987–1006.

[1990a] Undecidability and initial segments of the *wtt*-degrees below $\mathbf{0}'$, *Springer Lect. Not. Math.* 1432 (1990) 223–244.

Havel, I.M.

[1971] Weak complexity measures, *S.I.G.A.C.T. News* 3 (1971) 21–30.

Hay, L.

[1965] On creative sets and indices of partial recursive functions, *Trans. Am. Math. Soc.* 120 (1965) 359–367.

[1966] Isomorphism types of index sets of partial recursive functions, *Proc. Am. Math. Soc.* 17 (1966) 106.

[1969] Index sets of finite classes of recursively enumerable sets, *J. Symb. Log.* 34 (1969) 39–44.

[1972] A discrete chain of degrees of index sets, *J. Symb. Log.* 37 (1972) 139–149.

[1973] Index sets in $\mathbf{0}'$, *Alg. Log.* 12 (1973) 713–729, transl. 12 (1973) 408–416.

[1973a] Discrete ω-sequences of index sets, *Trans. Am. Math. Soc.* 183 (1973) 293–311.

[1973b] The classes of recursively enumarable subsets of a recursively enumerable set, *Pac. J. Math.* 46 (1973) 167–183.

[1973c] The halting problem relativized to complements, *Proc. Am. Math. Soc.* 41 (1973) 583–587.

[1974] A non-initial segment of index sets, *J. Symb. Log.* 39 (1974) 209–224.

[1974a] Index sets universal for differences of arithmetical sets, *Zeit. Math. Log. Grund. Math.* 20 (1974) 239–254.

[1975] Rice theorem for differences of recursively enumerable sets, *Can. J. Math.* 27 (1975) 352–365.

[1976] Boolean combinations of r.e. sets, *J. Symb. Log.* 41 (1976) 235–238.

Hay, L., Manaster, A.B., and Rosenstein, J.G.

[1975] Small recursive ordinals, many-one degrees, and the arithmetical difference hierarchy, *Ann. Math. Log.* 8 (1975) 297–343.

Heinermann, W.

[1961] *Untersuchungen über die Rekursionzahlen rekursiver Funktionen*, Ph.D. Thesis, Munich University, 1961.

Heller, H.

[1984] Relativized polynomial hierarchies extending two levels, *Math. Syst. Th.* 17

(1984) 71–84.

Helm, J.P., and Young, P.R.

[1971] On size vs. efficiency for programs admitting speed-ups, *J. Symb. Log.* 36 (1971) 21–27.

Hemachandra, L.A.

[1988] Structure of complexity classes: separations, collapses, and completeness, *Springer Lect. Not. Comp. Sci.* 324 (1988) 59–73.

[1989] The strong exponential hierarchy collapses, *J. Comp. Syst. Sci.* 39 (1989) 299–322.

Hemachandra, L.A., Ogiwara, M., and Watanabe, O.

[1992] How hard are sparse sets?, *Proc. Symp. Struct. Compl.* 7 (1992) 222–238.

Hempel, C.G.

[1945] Studies in the logic of confirmation, *Mind* 54 (1945).

Hennie, F.C.

[1965] One tape, off-line Turing machines computations, *Inf. Contr.* 8 (1965) 553–578.

Hennie, F.C., and Stearns, R.E.

[1966] Two tape simulation of multitape Turing machines, *J. Ass. Comp. Mach.* 13 (1966) 533–546.

Herbrand, J.

[1931] Sur la non-contradiction de l'Arithmétique, *J. Reine Angew. Math.* 166 (1931) 1–8, transl. in Van Heijenoort [1967], pp. 620–628.

Herman, G.T.

[1969] A new hierarchy of elementary functions, *Proc. Am. Math. Soc.* 20 (1969) 557–562.

[1971] The equivalence of different hierarchies of elementary functions, *Zeit. Math. Log. Grund. Math.* 17 (1971) 219–224.

Herrmann, E.

[1983] Orbits of hyperhypersimple sets and the lattice of Σ_3^0 sets, *J. Symb. Log.* 48 (1983) 693–699.

[1983a] *Major subsets of hypersimple sets and ideal families*, Ph.D. Thesis, Berlin University, 1983.

[1984] The undecidability of the elementary theory of the lattice of recursively enumerable sets, *Frege Conf.* 2 (1984) 66-72.

[1986] The index set $\{e : \mathcal{W}_e \equiv_1 X\}$, *J. Symb. Log.* 51 (1986) 110–116.

[1996] On the $\forall\exists$-theory of the factor lattice by the major subset relation, *Lond. Math. Soc. Lect. Not.* 224 (1996) 139–166.

Hesse, M.

[1976] *The structure of scientific inference*, 1974.

Hilbert, D.

[1926] Über das Unendliche, *Math. Ann.* 95 (1926) 161–190, transl. in Van Heijenoort [1967], pp. 367–392.

Hilbert, D., and Bernays, P.

[1939] *Grundlagen der Mathematik*, vol. II, Berlin, 1939.

Hinman, P.G.

[1969] Some applications of forcing to hierarchy problems in arithmetic, *Zeit. Math. Log. Grund. Math.* 15 (1969) 341–352.

Hirschfeld, J.

[1975] Models of arithmetic and recursive functions, *Isr. J. Math.* 20 (1975) 111–126.

Hodges, W., and Nies, A.

[1998] Noninterpretability of infinite linear orders, in *Logic Colloquium '95*, Makowski et al. eds., Springer, 1998, pp. 73–78.

Hodgson, B.R., and Kent, C.F.

[1983] A normal form for arithmetical representation of NP sets, *J. Comp. Syst. Sci.* 27 (1983) 378–388.

Homer, S.

[1985] Minimal polynomial degrees of nonrecursive sets, *Springer Lect. Not. Math.* 1141 (1985) 193–202.

[1986] On simple and creative sets in NP, *Theor. Comp. Sci.* 47 (1986) 169–180.

[1987] Minimal degrees for polynomial reducibilities, *J. Ass. Comp. Mach.* 34 (1987) 480–491.

Homer, S., and Long, T.J.

[1987] Honest polynomial degrees and P = NP, *Theor. Comp. Sci.* 51 (1987) 265–280.

Homer, S., and Longpré, L.

[1994] On reductions of NP sets to sparse sets, *J. Comp. Syst. Sci.* 48 (1994) 324–336.

Homer, S., and Maass, W.

[1983] Oracle-dependent properties of the lattice of NP sets, *Theor. Comp. Sci.* 24 (1983) 279–289.

Homer, S., and Selman, A.L.

[1989] Oracles for structural properties: the isomorphism problem and public-key cryptography, *Proc. Symp. Struct. Compl.* 4 (1989) 3–14.

Hopcroft, J.E., Paul, W., and Valiant, L.

[1977] On time versus space, *J. Ass. Comp. Mach.* 24 (1977) 332–337.

Hopcroft, J.E., and Ullman, J.D.

[1967] Nonerasing stack automata, *J. Comp. Syst. Sci.* 1 (1967) 166–186.

[1968] Relations between time and tape complexity, *J. Ass. Comp. Mach.* 15 (1968) 414–427.

[1969] Some results on tape bounded Turing machines, *J. Ass. Comp. Mach.* 16 (1969) 168–177.

[1979] *Introduction to Automata Theory, Languages and Computations*, Addison Wesley, 1979.

Howard, W.A.

[1972] A system of abstract constructive ordinals, *J. Symb. Log.* 37 (1972) 355–374.

Hume, D.

[1739] *A treatise of human nature*, 1739.

Ibarra, O.H.

[1971] Characterizations of some tape and time complexity classes of Turing machines in terms of multihead and auxiliary stack automata, *J. Comp. Syst. Sci.* 5 (1972) 608–612.

[1972] A note concerning nondeterministic tape complexity, *J. Ass. Comp. Mach.* 19 (1972) 608–612.

Ibarra, O.H., and Moran, S.

[1983] Some time-space trade-off results concerning single-tape and off-line Turing machines, *S.I.A.M. J. Comp.* 12 (1983) 388–394.

Immerman, N.

[1982] Relational queries computable in polynomial time, *Proc. Symp. Th. Comp.* 14

(1982) 147–152.
[1983] Languages which capture complexity classes, *Proc. Symp. Th. Comp.* 15 (1983) 347–354.
[1988] Nondeterministic space is closed under complementation, *S.I.A.M. J. Comp.* 17 (1988) 935–938.
[1989] Descriptive and computational complexity, *Proc. Symp. Appl. Math.* 38 (1989) 75–91.
[1999] *Descriptive complexity*, Springer Verlag, 1999.

Ingrassia, M.
[1981] *P-genericity for recursively enumerable sets*, Ph.D. Thesis, University of Illinois, Urbana, 1981.

Ischmuchametov, S.I.
[1982] On families of recursively enumerable sets, *Prob. Meth. Cyb.* 18 (1982) 46–53.
[1983] On the index sets of classes of differences (of r.e. sets), *Izv. Vyss. Uch. Zav. Mat.* 250 (1983) 78–79, transl. 27,3 (1983) 99–102.
[1985] Differences of recursively enumerable sets, *Izv. Vyss. Uch. Zav. Mat.* 279 (1985) 3–12, transl. 29,8 (1985) 1–13.

Jantke, K.
[1991] Monotonic and non-monotonic inductive inference, *New Gen. Comp.* 8 (1991) 349–360.
[1991a] Monotonic and non-monotonic inductive inference of functions and patterns, *Springer Lect. Not. Comp. Sci.* 543 (1991) 161–177.

Jantke, K., and Beik, H.
[1981] Combining postulates of naturalness in inductive inference, *J. Inf. Proc. Cyb.* 17 (1981) 465–484.

Jenner, B., Kirsig, B., and Lange, K.J.
[1989] The Logarithmic Alternation Hierarchy collapses: $A\Sigma_2^L = A\Pi_2^L$, *Inf. Comp.* 80 (1989) 269–288.

Jiang, Z.
[1993] The diamond lattice embedded into the *d*-r.e. degrees, *Science in China* 36 (1993) 803–811.

Jiang, Z., and Sui, Y.
[1995] The relative noncappability of the r.e. *wtt*-degrees, *Chin. J. Adv. Soft. Res.* 2 (1995) 381–388.

Jockusch, C.G.
[1969] Relationships between reducibilities, *Trans. Am. Math. Soc.* 142 (1969) 229–237.
[1969a] The degrees of hyperhyperimmune sets, *J. Symb. Log.* 34 (1969) 489–493.
[1972b] Degrees in which the recursive sets are uniformly recursive, *Can. J. Math.* 24 (1972) 1092–1099.
[1973] An application of Σ_4^0-determinacy to the degrees of unsolvability, *J. Symb. Log.* 38 (1973) 293–294.
[1977] Simple proofs of some theorems on high degrees of unsolvability, *Can. J. Math.* 29 (1977) 1072–1080.
[1981] Degrees of generic sets, *Lond. Math. Soc. Lect. Not.* 45 (1981) 110–139.
[1981a] Three easy constructions of recursively enumerable degrees, *Springer Lect. Not. Math.* 859 (1981) 83–91.
[1985] Genericity for recursively enumerable degrees, *Springer Lect. Not. Math.* 1141 (1985) 203–232.

Jockusch, C.G., Lerman, M., Soare, R.I., and Solovay, R.M.
[1989] Recursively enumerable sets modulo iterated jumps and extensions of Arslanov's

completeness criterion, *J. Symb. Log.* 54 (1989) 1288–1323.

Jockusch, C.G., and McLaughlin, T.G.

[1969] Countable retracing functions and Π^0_2 predicates, *Pac. J. Math.* 30 (1969) 67–93.

Jockusch, C.G., and Mohrherr, J.

[1985] Embedding the diamond lattice in the recursively enumerable truth-table degrees, *Proc. Am. Math. Soc.* 94 (1985) 123–128.

Jockusch, C.G., and Paterson, M.

[1976] Completely autoreducible degrees, *Zeit. Math. Log. Grund. Math.* 22 (1976) 571–575.

Jockusch, C.G., and Posner, D.

[1978] Double-jumps of minimal degrees, *J. Symb. Log.* 43 (1978) 715–724.

[1981] Automorphism bases for degrees of unsolvability, *Isr. J. Math.* 40 (1981) 150–164.

Jockusch, C.G., and Shore, R.A.

[1983] Pseudo-jump operators I: the r.e. case, *Trans. Am. Math. Soc.* 275 (1983) 599–609.

[1984] Pseudo-jump operators II: transfinite iterations, hierarchies and minimal covers, *J. Symb. Log.* 49 (1984) 1205–1236.

[1985] R.E.A. operators, r.e. degrees, and minimal covers, *Proc. Symp. Pure Appl. Math.* 42 (1985) 3–11.

Jockusch, C.G., and Soare, R.I.

[1970] Minimal covers and arithmetical sets, *Proc. Am. Math. Soc.* 25 (1970) 856–859.

Johnson, D.S.

[1990] A catalog of complexity classes, in *Handbook of Theoretical Computer Science*, Volume A, van Leeuwen ed., North Holland, 1990, pp. 67–161.

Johnson, N.

[1978] Classification of generalized index sets of open classes, *J. Symb. Log.* 43 (1978) 694–714.

Jones, N.D.

[1975] Space-bounded reducibility among combinatorial problems, *J. Comp. Syst. Sci.* 11 (1975) 68–85.

Jones, N.D., and Laaser, W.T.

[1976] Complete problems for deterministic polynomial time, *Theor. Comp. Sci.* 3 (1976) 105–118.

Jones, N.D., Lien, E., and Laaser, W.T.

[1976] New problems complete for nondeterministic log-space, *Math. Sys. Th.* 10 (1976) 1–17.

Jones, N.D., and Matijasevich, Y.

[1984] Register machine proof of the theorem on exponential diophantine representation of enumerable sets, *J. Symb. Log.* 49 (1984) 818–829.

Jones, N.D., and Selman, A.L.

[1974] Turing machines and the spectra of first-order formulas, *J. Symb. Log.* 29 (1974) 139–150.

Joseph, D., and Young, P.R.

[1981] A survey of some recent results on computational complexity in weak theories of arithmetic, *Springer Lect. Not. Comp. Sci.* 118 (1981) 46–60.

[1985] Some remarks on witness functions for nonpolynomial and noncomplete sets in NP, *Theor. Comp. Sci.* 39 (1985) 225–237.

Kaddah, D.

[1993] Infima in the d-r.e. degrees, *Ann. Pure Appl. Log.* 62 (1993) 207–263.

Kadin, J.

[1987] $\mathrm{P}^{\mathrm{NP}[\log n]}$ and sparse Turing-complete sets for NP, *Proc. Symp. Struct. Compl.* 2 (1987) 33–40.

[1988] The Polynomial Hierarchy collapses if the Boolean Hierarchy collapses, *S.I.A.M. J. Comp.* 17 (1988) 1263–1282.

Kallibekov, S.

[1971] On index sets of m-degrees, *Sib. Math. J.* 12 (1971) 1292–1300, transl. 12 (1971) 931–937.

[1971a] Index sets of degrees of unsolvability, *Alg. Log.* 10 (1971) 316–326, transl. 10 (1971) 198–204.

[1973] On degrees of recursively enumerable sets, *Sib. Math. J.* 14 (1973) 421–426, transl. 14 (1973) 290–293.

[1973a] On tt-degree of recursively enumerable sets, *Mat. Zam.* 14 (1973) 697–702, transl. 14 (1973) 958–961.

Kalmar, L.

[1943] Ein einfaches Beispiel für ein unentscheidbares arithmetisches Problem, *Mat. Fiz. Lapok* 50 (1943) 1–23.

Kanovich, M.I.

[1969] On the decision complexity of algorithms, *Dokl. Acad. Nauk* 186 (1969) 1008–1009, transl. 10 (1969) 700–701.

[1970a] On the decision complexity of a recursively enumerable set as a criterion for its completeness, *Dokl. Acad. Nauk* 194 (1970) 500–503, transl. 11 (1970) 1224–1228.

Kapron, B.M., and Cook, S.A.

[1996] A new characterization of type-2 feasibility, *S.I.A.M. J. Comp.* 25 (1996) 117–132.

Karp, R.M.

[1972] Reducibility among combinatorial problems, in *Complexity of Computer Computations*, Miller et al. eds., Plenum Press, 1972, pp. 85–103.

Karp, R.M., and Lipton, R.J.

[1980] Some connections between nonuniform and uniform complexity classes, *Proc. Symp. Th. Comp.* 12 (1980) 302–309.

Karp, R.M., and Ramachandran, V.

[1990] Parallel algorithms for shared-memory machines, in *Handbook of Theoretical Computer Science*, Volume A, van Leeuwen ed., North Holland, 1990, pp. 869–941.

Kasai, T., and Adachi, A.

[1980] A characterization of time complexity by simple loop programs, *J. Comp. Syst. Sci.* 20 (1980) 1–17.

Kazanovich, Y.B.

[1970] A classification of the primitive recursive functions with the help of Turing machines, *Probl. Kibern.* 22 (1970) 95–106.

Kent, C.F.

[1969] Reducing ordinal recursion, *Proc. Am. Math. Soc.* 22 (1969) 690–696.

Kent, C.F., and Hodgson, B.R.
[1982] An arithmetical characterization of NP, *Theor. Comp. Sci.* 21 (1982) 255–267.

Ketonen, J., and Solovay, R.M.
[1981] Rapidly growing Ramsey functions, *Ann. Math.* 113 (1981) 267–314.

Keynes, J.M.
[1921] *A treatise on probability*, 1921.

Khachian, L.G.
[1979] A polynomial algorithm for Linear Programming, *Dokl. Acad. Nauk* 244 (1979) 1093–1096, transl. 20 (1979) 191–194.

Kinber, E.B.
[1977] On a theory of inductive inference, *Springer Lect. Not. Comp. Sci.* 56 (1977) 435–440.

Kinber, E.B., and Zeugmann, T.
[1985] Inductive inference of almost everywhere correct programs by reliably working strategies, *J. Inf. Proc. Cyb.* 21 (1985) 91–100.

Kinber, J.
[1977] On *btt*-degrees of sets of minimal numbers in Gödel numberings, *Zeit. Math. Log. Grund. Math.* 23 (1977) 201–212.

Kino, A.
[1968] On provably recursive functions and ordinal recursive functions, *J. Math. Soc. Jap.* 20 (1968) 456–476.

Kintala, C.M.R., and Fischer, P.C.
[1980] Refining nondeterminism in relativized polynomial-time bounded computations, *S.I.A.M. J. Comp.* 9 (1980) 46–53.

Kirby, L.A., and Paris, J.B.
[1978] Σ_n-collection schemas in arithmetic, in *Logic Colloquium '77*, MacIntyre et al. eds., North Holland, 1978, pp. 199-209.
[1982] Accessible independence results for $\mathcal{PA}$, *Bull. Lond. Math. Soc.* 14 (1982) 285–293.

Kirsig, B., and Lange, K.J.
[1987] Separation with the Ruzzo, Simon and Tompa relativization implies DSPACE$[\log n] \neq$ NSPACE$[\log n]$, *Inf. Proc. Lett.* 25 (1987) 13–15.

Klee, V., and Minty, G.J.
[1972] How good is the Simplex Algorithm?, in *Inequalities III*, Shisha ed., Academic Press, 1972, pp. 159–175.

Kleene, S.K.
[1936] General recursive functions of natural numbers, *Math. Ann.* 112 (1936) 727–742, also in Davis [1965], pp. 237–252.
[1947] Review of Post [1946], *J. Symb. Log.* 12 (1947) 28.
[1952] *Introduction to metamathematics*, North Holland, 1952.
[1956] Representation of events in nerve nets and finite automata, in *Automata studies*, Shannon et al. eds., Princeton University Press, 1956, pp. 3–42.
[1958] Extension of an effectively generated class of functions by enumeration, *Colloq. Math.* 6 (1958) 67–78.

Kleene, S.K., and Post, E.L.
[1954] The uppersemilattice of degrees of recursive unsolvability, *Ann. Math.* 59 (1954) 379–407.

Klette, R.

[1976] Erkennung allgemein-rekursiver Funktionen, *J. Inf. Proc. Cyb.* 12 (1976) 227–243.

Klette, R., and Wiehagen, R.

[1980] Research in the theory of Inductive Inference by GDR mathematicians - A survey, *Inf. Sci.* 22 (1980) 149–169.

Kloss, B.M.

[1964] The definition of complexity of algorithms, *Dokl. Acad. Nauk* 157 (1964) 38–40, transl. 5 (1964) 880–882.

Knight, J.

[1990] A metatheorem for constructions by finitely many workers, *J. Symb. Log.* 55 (1990) 787–804.

Ko, K.I.

[1985] On some natural complete operators, *Theor. Comp. Sci.* 37 (1985) 1–30.

[1988] Relativized Polynomial Time Hierarchies having exactly k levels, *Proc. Symp. Th. Comp.* 20 (1988) 245–253.

[1989] Distinguishing conjunctive and disjunctive reducibilities by sparse sets, *Inf. Compl.* 81 (1989) 62–87.

[1991] Separating the Low and High Hierarchies by oracles, *Inf. Comp.* 90 (1991) 156–177.

Ko, K.I., Long, T., and Du, D.

[1987] A note on one-way functions and polynomial time isomorphisms, *Theor. Comp. Sci.* 47 (1987) 263–276.

Ko, K.I., and Moore, D.

[1981] Completeness, approximation and density, *S.I.A.M. J. Comp.* 10 (1981) 787–796.

Ko, K.I., and Schöning, U.

[1985] On circuit-size complexity and the Low Hierarchy in NP, *S.I.A.M. J. Comp.* 14 (1985) 41–51.

Kobayashi, K.

[1985] On proving time constructibility of functions, *Theor. Comp. Sci.* 35 (1985) 215–225.

Kobzev, G.N.

[1973] *btt*-reducibility, *Alg. Log.* 12 (1973) 190–204, transl. 12 (1973) 107–115.

[1973a] *btt*-reducibility II, *Alg. Log.* 12 (1973) 433–444, transl. 12 (1973) 242–248.

[1974] On the complete *btt*-degree, *Alg. Log.* 13 (1974) 22-25, transl. 13 (1974) 10–12.

[1975] On r-separable sets, in *Studies of Mathematical Logic and the Theory of Algorithms*, Tbilisi, 1975, pp. 19–30.

[1976] Relationships between recursively enumerable *tt*- and *w*-degrees, *Bull. Acad. Sci. Georg.* 84 (1976) 585–587.

[1977] On recursively enumerable *bw*-degrees, *Mat. Zam.* 21 (1977) 839–846, transl. 21 (1977) 473–477.

[1978] On *tt*-degrees of recursively enumerable Turing degrees, *Mat. Sborn.* 106 (1978) 507–514, transl. 35 (1979) 173–180.

Kontostathis, K.

[1991] Topological framework for non-priority, *Zeit. Math. Log. Grund. Math.* 37 (1991) 495–500.

[1992] Topological framework for finite injury, *Zeit. Math. Log. Grund. Math.* 38 (1992) 189–195.

[199?] Topological framework for infinite injury, *to appear*.

Kowalczyk, W.

[1984] Some connections between presentability of complexity classes and the power of formal systems of reasoning, *Springer Lect. Not. Comp. Sci.* 176 (1984) 364–369.

Kozen, D.

[1980] Indexing of subrecursive classes, *Theor. Comp. Sci.* 11 (1980) 277–301.

Kozmiadi, V.A.

[1970] On a generalization of finite automata generating a hierarchy analogous to the Grzegorczyk classification of the primitive recursive functions, *Probl. Kibern.* 23 (1970) 127–170.

Kozmiadi, V.A., and Marchenkov, S.S.

[1969] On multihead automata, *Probl. Kibern.* 21 (1969) 129–160.

Kreisel, G.

[1951] On the interpretation of nonfinitist proofs, I, *J. Symb. Log.* 16 (1951) 241–267.

[1952] On the interpretation of nonfinitist proofs, II, *J. Symb. Log.* 17 (1952) 43–58.

[1958] A remark on free choice sequences and the topological completeness proofs, *J. Symb. Log.* 23 (1958) 369–388.

[1958a] Mathematical significance of consistency proofs, *J. Symb. Log.* 23 (1958) 155–182.

[1959] Interpretation of analysis by means of constructive functionals of finite types, in *Constructivity in mathematics*, Heyting ed., North Holland, 1959, pp. 101–128.

[1960a] Non uniqueness results for transfinite progressions, *Bull. Acad. Pol. Sci.* 5 (1960) 287–290.

Kreisel, G., Shoenfield, J.R., and Wang, H.

[1960] Number theoretic concepts and recursive well-orderings, *Arch. Math. Log. Grund.* 5 (1960) 42–64.

Kripke, S.

[1965] Semantical analysis of intuitionistic logic, in *Formal systems and recursive function Theory*, Crossley at al. eds., North Holland, 1965, pp. 92–130.

Kristiansen, L.

[199?] A jump operator on honest subrecursive degrees, *to appear*.

Kučera, A.

[1986] An alternative, priority-free solution to Post's Problem, *Springer Lect. Not. Comp. Sci.* 233 (1986) 493–500.

[1988] On the role of $\mathbf{0}'$ in recursion theory, in *Logic Colloquium '86*, Drake et al. eds., North Holland, 1988, pp. 133-141.

[1989] On the use of diagonally nonrecursive functions, in *Logic Colloquium '87*, Ebbinghaus et al. eds., North Holland, 1989, pp. 219–239.

Kugel, P.

[1977] Induction, pure and simple, *Inf. Contr.* 35 (1977) 276–336.

Kumabe, M.

[1990] A 1-generic degree which bounds a minimal degree, *J. Symb. Log.* 55 (1990) 733–743.

[1993] Generic degrees are complemented, *Ann. Pure Appl. Log.* 59 (1993) 257–272.

[1993a] Every n-generic degree is a minimal cover of an n-generic degree, *J. Symb. Log.* 58 (1993) 219–231.

[1996] Degrees of generic sets, *Lond. Math. Soc. Lect. Not.* 224 (1996) 167–183.

[199?] On the structure of 1-generic degrees below $\mathbf{0}'$, *to appear*.

Kummer, M.

[1989] Numberings in $R_1 \cup F$, *Springer Lect. Not. Comp. Sci.* 385 (1989) 166–186.

[1990] An easy priority-free proof of a theorem of Friedberg, *Theor. Comp. Sci.* 74 (1990) 249–251.

[1993] Degrees of unsolvability in abstract complexity theory, in *Complexity Theory. Current research*, Ambos-Spies ed., Cambridge University Press, 1993, pp. 227–243.

Kummer, M., and Stephan, F.

[1996] On the structure of degrees of inferability, *J. Comp. Syst. Sci.* 52 (1996) 214–238.

Kunen, K.

[1980] *Set Theory*, North Holland, 1980.

Kuroda, S.Y.

[1964] Classes of languages and linear-bounded automata, *Inf. Contr.* 7 (1964) 207–223.

Kurtz, S.A.

[1981] *Randomness and genericity in the degrees of unsolvability*, Ph.D. Thesis, University of Illinois, Urbana, 1981.

[1983] Notions of weak genericity, *J. Symb. Log.* 48 (1983) 764–770.

[1983a] On the random oracle hypothesis, *Inf. Contr.* 57 (1983) 40–47.

[1985] Sparse sets in NP − P: relativizations, *S.I.A.M. J. Comp.* 14 (1985) 113–119.

Kurtz, S.A., Mahaney, S.R., and Royer, J.

[1989] The isomorphism conjecture fails relative to a random oracle, *Proc. Symp. Th. Comp.* 21 (1989) 157–166.

[1990] The structure of complete degrees, in *Complexity Theory Retrospective*, Selman ed., Springer, 1990, pp. 108–146.

Kusmina, T.M.

[1981] Structure of m-degrees of index sets of families of partial recursive functions, *Alg. Log.* 20 (1981) 55–68, transl. 20 (1981) 37–48.

Kuznekov, A.V., and Trakhtenbrot, B.A.

[1955] Investigation of partially recursive operators by means of the theory of Baire space, *Dokl. Acad. Nauk* 105 (1955) 897–900.

Lachlan, A.H.

[1962] Multiple recursion, *Zeit. Math. Log. Grund. Math.* 8 (1962) 81–107.

[1965] Some notions of reducibility and productiveness, *Zeit. Math. Log. Grund. Math.* 11 (1965) 17–44.

[1965c] On a problem of G.E. Sacks, *Proc. Am. Math. Soc.* 16 (1965) 972–979.

[1966] A note on universal sets, *J. Symb. Log.* 31 (1966) 573–574.

[1966a] The impossibility of finding relative complements for recursively enumerable degrees, *J. Symb. Log.* 31 (1966) 434–454.

[1966b] Lower bounds for pairs of recursively enumerable degrees, *Proc. Lond. Math. Soc.* 16 (1966) 537–569.

[1967a] The priority method I, *Zeit. Math. Log. Grund. Math.* 13 (1967) 1–10.

[1968b] On the lattice of recursively enumerable sets, *Trans. Am. Math. Soc.* 130 (1968) 1–37.

[1968c] The elementary theory of recursively enumerable sets, *Duke Math. J.* 35 (1968) 123–146.

[1968d] Degrees of recursively enumerable sets which have no maximal superset, *J. Symb. Log.* 33 (1968) 431–443.

[1969] Initial segments of one-one degrees, *Pac. J. Math.* 29 (1969) 351–366.

[1970a] On some games which are relevant to the theory of recursively enumerable sets, *Ann. Math.* 91 (1970) 291–310.

[1972] Two theorems on many-one degrees of recursively enumerable sets, *Alg. Log.* 11 (1972) 216–229, transl. 11 (1972) 127–132.

[1972a] Recursively enumerable many-one degrees, *Alg. Log.* 11 (1972) 326–358, transl. 11 (1972) 186–202.

[1972b] Embedding nondistributive lattices in the recursively enumerable degrees, *Springer Lect. Not. Math.* 255 (1972) 149–177.

[1973] The priority method for the construction of recursively enumerable sets, *Springer Lect. Not. Math.* 337 (1973) 299–310.

[1975] *wtt*-complete sets are not necessarily *tt*-complete, *Proc. Am. Math. Soc.* 48 (1975) 429–434.

[1975a] A recursively enumerable degree which will not split over all lesser ones, *Ann. Math. Log.* 9 (1975) 307–365.

[1975b] Uniform enumeration operators, *J. Symb. Log.* 40 (1975) 401–409.

[1979] Bounding minimal pairs, *J. Symb. Log.* 44 (1979) 626–642.

[1980] Decomposition of recursively enumerable degrees, *Proc. Am. Math. Soc.* 79 (1980) 629–634.

Lachlan, A.H., and Shore, R.A.

[1992] The n-r.e.a. enumeration degrees are dense, *Arch. Math. Log.* 31 (1992) 227–285.

Lachlan, A.H., and Soare, R.I.

[1980] Not every finite lattice is embeddable in the recursively enumerable degrees, *Adv. Math.* 37 (1980) 74–82.

Ladner, R.E.

[1973] Mitotic recursively enumerable sets, *J. Symb. Log.* 38 (1973) 199–211.

[1973a] A completely mitotic nonrecursive recursively enumerable degree, *Trans. Am. Math. Soc.* 184 (1973) 479–507.

[1975] On the structure of polynomial time reducibilities, *J. Ass. Comp. Mach.* 22 (1975) 155–171.

[1981] Complexity Theory with emphasis on the complexity of logical theories, *Lond. Math. Soc. Lect. Not.* 45 (1981) 286–319.

Ladner, R.E., Lipton R.J., and Stockmeyer, L.J.

[1978] Alternating pushdown automata, *Proc. Symp. Found. Comp. Sci.* 19 (1978) 92–106.

Ladner, R.E., and Lynch, N.A.

[1976] Relativization of questions about log-space computability, *Math. Syst. Th.* 10 (1976) 19–32.

Ladner, R.E., Lynch, N.A., and Selman, A.L.

[1975] A comparison of polynomial time reducibilities, *Theor. Comp. Sci.* 1 (1975) 103–123.

Ladner, R.E., and Sasso, L.P.

[1975] The weak truth-table degrees of recursively enumerable sets, *Ann. Math. Log.* 8 (1975) 429–448.

LaForte, G.

[1995] The isolated d-r.e. degrees are dense in the r.e. degrees, *Math. Log. Quart.* 42 (1995) 83–103.

Lagemann, J.T.

[1971] *Embedding theorems in the reducibility ordering of partial degrees*, Ph.D. Thesis, M.I.T., 1971.

Lamé, G.

[1884] Note sur la limite du nombre des divisions dans la récherche du plus grand commun diviseur entre deux nombres entiers, *Compt. Rend. Acad. Sci.* 19 (1884) 867–870.

Landweber, L.H., Lipton, R.J., and Robertson, E.L.

[1981] On the structure of sets in NP and other complexity classes, *Theor. Comp. Sci.* 15 (1981) 181–200.

Landweber, L.H., and Robertson, E.L.

[1972] Recursive properties of abstract complexity classes, *J. Ass. Comp. Mach.* 19 (1972) 296–308.

Landweber, P.S.

[1963] Three theorems on phrase structure grammars of type 1, *Inf. Contr.* 6 (1963) 131–136.

Lautemann, C.

[1983] BPP and the Polynomial Hierarchy, *Inf. Proc. Lett.* 17 (1983) 215–217.

Leivant, D.

[1989] Descriptive characterizations of computational complexity, *J. Comp. Syst. Sci.* 39 (1989) 51–83.

Lempp, S.

[1987] Hyperarithmetical index sets in recursion theory, *Trans. Am. Math. Soc.* 303 (1987) 559–583.

Lempp, S., and Lerman, M.

[1990] Priority arguments using iterated trees of strategies, *Springer Lect. Not. Math.* 1432 (1990) 277–296.

[1995] General framework for priority arguments, *Bull. Symb. Log.* 1 (1995) 189–201.

[1997] A finite lattice without critical triple that cannot be embedded into the recursively enumerable Turing degrees, *Ann. Pure Appl. Log.* 87 (1997) 167–185.

[1997a] Iterated trees of strategies and priority arguments, *Arch. Math. Log.* 36 (1997) 297–312.

Lempp, S., and Nies, A.

[1995] Undecidability of the 4-quantifier theory for the recursively enumerable Turing and *wtt*-degrees, *J. Symb. Log.* 60 (1995) 1118–1135.

Lempp, S., Nies, A., and Slaman, T.

[1998] The Π_3-theory of r.e. Turing degrees is undecidable, *Trans. Am. Math. Soc.* 350 (1998) 2719–2736.

Lempp, S., and Sorbi, A.

[199?] Embedding finite lattices into the enumeration degrees of Σ_2^0 sets, *to appear*.

Lerman, M.

[1969] Some nondistributive lattices as initial segments of the degrees of unsolvability, *J. Symb. Log.* 34 (1969) 85–98.

[1970] Turing degrees and many-one degrees of maximal sets, *J. Symb. Log.* 35 (1970) 29–40.

[1970a] Recursive functions modulo co-r-maximal sets, *Trans. Am. Math. Soc.* 148 (1970) 429–444.

[1971a] Some theorems on r-maximal sets and major subsets of recursively enumerable sets, *J. Symb. Log.* 36 (1971) 193–215.

[1973] Admissible ordinals and priority arguments, *Springer Lect. Not. Math.* 337 (1973) 311-344.

[1974] Least upper bounds for minimal pairs of α-r.e. α-degrees, *J. Symb. Log.* 39 (1974) 49–56.

[1974a] Maximal α-r.e. sets, *Trans. Am. Math. Soc.* 188 (1974) 341–386.

[1976] Congruence relations, filters, ideals, and definability in lattices of α-recursively enumerable sets, *J. Symb. Log.* 41 (1976) 405–418.

[1983] *The degrees of unsolvability*, Springer, 1983.

[1985] On the ordering of classes in High-Low hierarchies, *Springer Lect. Not. Math.* 1141 (1985) 260–270.

[1986] Degrees which do not bound minimal degrees, *Ann. Pure Appl. Log.* 30 (1986) 249–276.

[1996] Embeddings into the recursively enumerable degrees, *Lond. Math. Soc. Lect. Not.* 224 (1996) 185–204.

[1998] A necessary and sufficient condition for embedding ranked finite partial lattices into the computably enumerable degrees, *Ann. Pure Appl. Log.* 94 (1998) 143–180.

[199?] A necessary and sufficient condition for embedding principally decomposable finite lattices into the computably enumerable degrees, *Ann. Pure Appl. Log.*, *to appear*.

Lerman, M., and Shore, R.A.

[1988] Decidability and invariant classes for degree structures, *Trans. Am. Math. Soc.* 310 (1988) 669–692.

Lerman, M., Shore, R.A., and Soare, R.I.

[1978] r-maximal major subsets, *Isr. J. Math.* 31 (1978) 1–18.

[1984] The elementary theory of the recursively enumerable degrees is not $\aleph_0$-categorical, *Adv. Math.* 53 (1984) 301–320.

Lerman, M., and Simpson, S.G.

[1973] Maximal sets in α-recursion theory, *Isr. J. Math.* 4 (1973) 236–247.

Lerman, M., and Soare, R.I.

[1980] A decidable fragment of the elementary theory of the lattice of recursively enumerable sets, *Trans. Am. Math. Soc.* 257 (1980) 1–37.

[1980a] d-simple sets, small sets and degree classes, *Pac. J. Math.* 87 (1980) 135–155.

Levin, L.A.

[1973] On storage capacity for algorithms, *Dokl. Acad. Nauk* 212 (1973) 804–805, transl. 14 (1973) 1464–1466.

[1973a] Universal sorting problems, *Probl. Pered. Inf.* 9 (1973) 115–116, transl. 9 (1973) 265–266.

[1974] Complexity of computation of computable functions, in *Complexity of Computations and Algorithms*, Kozmiadi et al. eds., Moscow, 1974.

Levy, A.

[1979] *Basic Set Theory*, Springer Verlag, 1979.

Lewis, F.D.

[1971] Classes of recursive functions and their index sets, *Zeit. Math. Log. Grund. Math.* 17 (1971) 291–294.

[1971a] The enumerability and invariance of complexity classes, *J. Comp. Syst. Sci.* 5 (1971) 286–303.

Lewis, H.R., and Papadimitriou, C.H.

[1981] *Elements of the theory of computation*, Prentice Hall, 1981.

Li, A.

[199?] External center theorem of the recursively enumerable degrees, *to appear*.

Li, A., and Yang, D.

[1998] Bounding minimal degrees by computably enumerable degrees, *J. Symb. Log.* 63 (1998) 1319–1347.

Li, A., and Yi, X.
[1999] Cupping the recursively enumerable degrees by d-r.e. degrees, *Proc. Lond. Math. Soc.* 78 (1999) 1–21.

Li, M.
[1985] *Lower bounds in computational complexity*, Ph.D. Thesis, Cornell University, 1985.

Li, M., and Vitányi, P.
[1988] Tape versus queue and stacks: the lower bounds, *Inf. Comp.* 78 (1988) 56–85.
[1993] *An introduction to Kolmogorov Complexity and applications*, Springer Verlag, 1993.

Lindner, R., and Werner, G.
[1976] Eine vergleichende Analyse von Stopstrategien für allgemein-rekursive Prognosen, *J. Inf. Proc. Cyb.* 12 (1976) 275– 280.

Lischke, G.
[1975] Über die Erfüllung gewisser Erhaltungssätze durch Kompliziertheitsmasse, *Zeit. Math. Log. Grund. Math.* 21 (1975) 159–166.
[1976] Natürliche Kompliziertheitsmasse und Erhaltungssätze I, *Zeit. Math. Log. Grund. Math.* 22 (1976) 413–418.
[1977] Natürliche Kompliziertheitsmasse und Erhaltungssätze II, *Zeit. Math. Log. Grund. Math.* 23 (1977) 193–200.
[1986] Oracle constructions to prove all possible relationships between relativizations of P, NP, EL, NEL, EP and NEP, *Zeit. Math. Log. Grund. Math.* 32 (1986) 257–270.
[1987] Relativizations of NP and EL, strongly separating, and sparse sets, *J. Inf. Proc. Cyb.* 23 (1987) 99–112.
[1989] The quality of separation between NP and Exponential Time; reducing the cases, *J. Inf. Proc. Cyb.* 25 (1989) 529–534.
[1990] Impossibilities and possibilities of weak separation between NP and Exponential Time, *Proc. Symp. Struct. Compl.* 5 (1990) 245–253.

Liskiewicz, M., and Lorys, K.
[1989] Some time-space bounds for one-tape deterministic Turing machines, *Springer Lect. Not. Comp. Sci.* 380 (1989) 297–307.

Liu, S.C.
[1960] An enumeration of the primitive recursive functions without repetitions, *Tohoku Math. J.* 12 (1960) 400–403.
[1960a] A theorem on general recursive functions, *Proc. Am. Math. Soc.* 11 (1960) 184–187.

Löb, M.H., and Wainer, S.S.
[1970] Hierarchies of number theoretic functions, I, *Arch. Math. Log.* 13 (1970) 39–51.
[1970a] Hierarchies of number theoretic functions, II, *Arch. Math. Log.* 13 (1970) 97–113.
[1971] Hierarchies of number theoretic functions. A correction, *Arch. Math. Log.* 14 (1971) 198–199.

Long, T.
[1982] A note on sparse oracles for NP, *J. Comp. Syst. Sci.* 24 (1982) 224–232.
[1982a] Strong nondeterministic polynomial time reducibilities, *Theor. Comp. Sci.* 21 (1982) 1–25.
[1985] On restricting the size of oracles compared with restricting the access to the oracle, *S.I.A.M. J. Comp.* 14 (1985) 585–597. Corrections, *ibidem*, 17 (1988) 628.

Long, T., and Selman, A.L.

[1986] Relativizing complexity classes with sparse oracles, *J. Ass. Comp. Mach.* 33 (1986) 618–628.

Loveland, D.W.

[1969] A variant of the Kolmogorov concept of complexity, *Inf. Contr.* 15 (1969) 510–526.

Lund, C., Fortnow, L., Karloff, H., and Nisan, N.

[1990] Algebraic methods for interactive proofs, *Proc. Symp. Found. Comp. Sci.* 31 (1990) 1–10.

Lutz, J.H., and Mayordomo, E.

[1996] Cook versus Karp-Levin: separating completeness notions if NP is not small, *Theor. Comp. Sci.* 164 (1996) 141–163.

Lynch, N.A.

[1975] Helping: several formulations, *J. Symb. Log.* 40 (1975) 555–566.

[1975a] On reducibility to complex or sparse sets, *J. Ass. Comp. Mach.* 22 (1975) 341–345.

[1978] Log-space machines with multiple oracle tapes, *Theor. Comp. Sci.* 6 (1978) 25–39.

Maass, W.

[1978] High α-recursively enumerable degrees, in *Generalized Recursion Theory II*, Fenstad et al. eds., North Holland, 1978, pp. 239–269.

[1982] Recursively enumerable generic sets, *J. Symb. Log.* 47 (1982) 809–823.

[1983] Characterization of recursively enumerable supersets effectively isomorphic to all recursively enumerable sets, *Trans. Am. Math. Soc.* 279 (1983) 311–336.

[1984] On the orbits of hyperhypersimple sets, *J. Symb. Log.* 49 (1984) 51–62.

[1985] Variations on promptly simple sets, *J. Symb. Log.* 50 (1985) 138–148.

[1985a] Major subsets and automorphisms of recursively enumerable sets, *Proc. Symp. Pure Appl. Math.* 42 (1985) 21–32.

[1985b] Combinatorial lower bounds for deterministic and nondeterministic one-tape Turing machines, *Trans. Am. Math. Soc.* 292 (1985) 675–693.

Maass, W., Shore, R.A., and Stob, M.

[1981] Splitting properties and jump classes, *Isr. J. Math.* 39 (1981) 210–224.

Maass, W., and Stob, M.

[1983] The intervals of the lattice of r.e. sets determined by major subsets, *Ann. Math. Log.* 24 (1983) 189–212.

Machtey, M.

[1972] Augmented loop languages and classes of computable functions, *J. Comp. Syst. Sci.* 6 (1972) 603–624.

[1974] The honest subrecursive classes are a lattice, *Inf. Contr.* 24 (1974) 247–263.

[1975] On the density of honest subrecursive classes, *J. Comp. Syst. Sci.* 10 (1975) 183–199.

[1975a] Heping and meet of pairs of honest subrecursive classes, *Inf. Contr.* 28 (1975) 76–89.

[1976] Minimal pairs of polynomial degrees with subexponential complexity, *Theor. Comp. Sci.* 2 (1976) 73–76.

MacIntyre, J.M.

[1977] Transfinite extensions of Friedberg's completeness theorem, *J. Symb. Log.* 42 (1977) 1–10.

Mahaney, S.R.

[1982] Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis,

J. Comp. Syst. Sci. 25 (1982) 130–143.
[1986] Sparse sets and reducibilities, in *Studies in Complexity Theory*, Book ed., John Wiley and Sons, 1986, pp. 63–118.
[1989] The isomorphism conjecture and sparse sets, *Proc. Symp. Appl. Math.* 38 (1989) 18–46.

Mahaney, S.R., and Young, P.R.
[1985] Reductions among polynomial isomorphism types, *Theor. Comp. Sci.* 39 (1985) 207–224.

Manaster, A.B.
[1971] Some contrasts between degrees and the arithmetical hierarchy, *J. Symb. Log.* 36 (1971) 301–304.

Marandzjan, G.B.
[1977] C-degrees of the sets of minimal indices of algorithms, *Izv. Acad. Nauk Armen.* 12 (1977) 130–137.
[1979] On the sets of minimal indices of partial recursive functions, *Springer Lect. Not. Comp. Sci.* 74 (1979) 372–374.

Marchenkov, S.S.
[1969] Elimination of the recursion schema in Grzegorczyk's $\mathcal{E}_2$ class, *Mat. Zam.* 5 (1969) 561–568, transl. 5 (1969) 336–340.
[1970] Multiple recursion bounded in the class of primitive recursive functions, *Kibern.* 6 (1970) 53–59.
[1972] On bounded recursion, *Math. Balk.* 2 (1972) 124–142.
[1975] The existence of recursively enumerable minimal tt-degrees, *Alg. Log.* 14 (1975) 422–429, transl. 14 (1975) 257–261.
[1975a] On Skolem-elementary functions, *Mat. Zam.* 17 (1975) 133–141, transl. 17 (1975) 79–83.
[1976] One class of partial sets, *Mat. Zam.* 20 (1976) 473–478, transl. 20 (1976) 823–825.
[1976a] On the comparison of the uppersemilattice of r.e. m-degrees and tt-degrees, *Mat. Zam.* 20 (1976) 19–26, transl. 20 (1976) 567–570.
[1976b] Tabular powers of maximal sets, *Mat. Zam.* 20 (1976) 373–381, transl. 20 (1976) 766–770.
[1977] On recursively enumerable minimal btt-degrees, *Mat. Sborn.* 103 (1977) 550–562, transl. 32 (1977) 477–487.
[1980] One base for composition of the class of Kalmar elementary functions, *Mat. Zam.* 27 (1980) 321–331, transl. 27 (1980) 161–166.

Marquez, I.
[1975] On speedability of recursively enumerable sets, *Zeit. Math. Log. Grund. Math.* 21 (1975) 199–214.
[1975a] On degrees of unsolvability and complexity properties, *J. Symb. Log.* 40 (1975) 529–540.

Martin, D.A.
[1963] A theorem on hyperhypersimple sets, *J. Symb. Log.* 28 (1963) 273–278.
[1966a] Classes of recursively enumerable sets and degrees of unsolvability, *Zeit. Math. Log. Grund. Math.* 12 (1966) 295–310.
[1966b] On a question of G.E. Sacks, *J. Symb. Log.* 31 (1966) 66–69.

Matijasevich, Y.
[1970] Enumerable sets are diophantine, *Dokl. Acad. Nauk* 191 (1970) 279–282, transl. 11 (1970) 354–357.

McCulloch, W., and Pitts, W.
[1943] A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.* 5 (1943) 115–133.

McEvoy, K.
[1985] Jumps of quasi-minimal enumeration degrees, *J. Symb. Log.* 50 (1985) 939–848.

McEvoy, K., and Cooper, S.B.
[1985] On minimal pairs of enumeration degrees, *J. Symb. Log.* 50 (1985) 983–1001.

McLaughlin, T.G.
[1965a] Strong reducibility on hypersimple sets, *Notre Dame J. Form. Log.* 6 (1965) 229–234.
[1971] The family of all recursively enumerable classes of finite sets, *Trans. Am. Math. Soc.* 155 (1971) 127–136.

Medvedev, Y.T.
[1955a] Degrees of difficulty of the mass problem, *Dokl. Acad. Nauk* 104 (1955) 501–504.

Melhorn, K.
[1976] Polynomial and abstract subrecursive classes, *J. Comp. Syst. Sci.* 12 (1976) 147–178.

Merkle, W.
[1996] *A generalized account of resource bounded reducibilities*, Ph.D. Thesis, Heidelberg University, 1996.

Merkle, W., and Stephan, F.
[1996] Trees and learning, *Proc. Conf. Comp. Learn. Th.* 9 (1996) 270–279.

Meyer, A.R.
[1965] Depth of nesting and the Grzegorczyk hierarchy, *Not. Am. Math. Soc.* 12 (1965) 342.
[1972] Program size in restricted programming languages, *Inf. Contr.* 21 (1972) 382–394.

Meyer, A.R., and Bagchi, A.
[1972] Program size and economy of description, *Proc. Symp. Th. Comp.* 4 (1972) 183–186.

Meyer, A.R., and Fischer, M.J.
[1972] Relatively complex recursive sets, *J. Symb. Log.* 37 (1972) 55–68.

Meyer, A.R., and Fischer, P.C.
[1972] Computational speed-up by effective operators, *J. Symb. Log.* 37 (1972) 55–68.

Meyer, A.R., and McCreight, E.M.
[1969] Classes of computable functions defined by bounds of computations, *Proc. Symp. Th. Comp.* 1 (1969) 79–88.

Meyer, A.R., and Ritchie, D.M.
[1967] The complexity of loop programs, *Proc. Ass. Comp. Mach. Conf.* 22 (1967) 465–469.
[1972] A classification of the recursive functions, *Zeit. Math. Log. Grund. Math.* 18 (1972) 71–82.

Meyer, A.R., and Stockmeyer, L.J.
[1972] The equivalence problem for regular expressions with squaring requires exponential time, *Proc. Symp. Found. Comp. Sci.* 13 (1972) 125–129.

Meyer, A.R., and Winklmann, K.

[1979] The fundamental theorem of complexity theory, *Found. Comp. Sci.* 3 (1979) 97–112.

Mill, J.S.

[1872] *A system of logic*, 1872.

Miller, D.P.

[1981] *The relationship between the structure and degrees of recursively enumerable sets*, Ph.D. Thesis, University of Chicago, 1981.

[1981a] High recursively enumerable degrees and the anticupping property, *Springer Lect. Not. Math.* 859 (1981) 230–245.

Miller, D.P., and Remmel, J.B.

[1984] Effectively nowhere simple sets, *J. Symb. Log.* 49 (1984) 129–136.

Miller, W., and Martin, D.A.

[1968] The degrees of hyperimmune sets, *Zeit. Math. Log. Grund. Math.* 14 (1968) 159–166.

Minicozzi, E.

[1976] Some natural properties of strong-identification in inductive inference, *Theor. Comp. Sci.* 2 (1976) 345–360.

Minsky, M.L.

[1967] *Computation: finite and infinite machines*, Prentice Hall, 1967.

Mints, G.E.

[1973] Quantifier-free and one-quantifier systems, *J. Sov. Math.* 1 (1973) 71–84.

Moh, S.K.

[1956] On the explicit form of general recursive functions, *Acta Mat. Sinica* (1956) 548–564.

Mohrherr, J.

[1983] Kleene index sets and functional m-degrees, *J. Symb. Log.* 48 (1983) 829–840.

Moll, R.

[1976] An operator embedding theorem for complexity classes of recursive functions, *Theor. Comp. Sci.* 1 (1976) 193–198.

Moll, R., and Meyer, A.R.

[1974] Honest bounds for complexity classes of recursive functions, *J. Symb. Log.* 39 (1974) 127–138.

Möllerfeld, M., and Weiermann, A.

[199?] A uniform approach to $\prec$-recursion, *to appear*.

Monien, B.

[1974] Characterization of time-bounded computations by limited primitive recursion, *Springer Lect. Not. Comp. Sci.* 14 (1974) 280–293.

[1975] About the deterministic simulation of nondeterministic $\log n$-tape bounded Turing machines, *Springer Lect. Not. Comp. Sci.* 33 (1975) 118–126.

[1977] A recursive and grammatical characterization of the exponential time languages, *Theor. Comp. Sci.* 3 (1977) 61–74.

Monien, B., and Sudborough, I.H.

[1982] On eliminating non-determinism from Turing machines which use less than logarithmic work tape space, *Theor. Comp. Sci.* 21 (1982) 237–253.

Monk, J.D.

[1969] *Introduction to Set Theory*, McGraw Hill, 1969.

Monk, J.D., and Bonnett, R.

[1989] *Handbook of Boolean algebras*, North Holland, 1989.

Moore, G.H.

[1988] The origins of forcing, in *Logic Colloquium '86*, Drake et al. eds., North Holland, 1988, pp. 143–173.

Moran, S.

[1981] Some results on relativized deterministic and non deterministic time hierarchies, *J. Comp. Syst. Sci.* 22 (1981) 1–8.

Morley, M.D., and Soare, R.I.

[1975] Boolean algebras, splitting theorems and Δ_2^0 sets, *Fund. Math.* 90 (1975) 45–52.

Morris, P.H.

[1974] *Complexity theoretic properties of recursively enumerable sets*, Ph.D. Thesis, University of California, Irvine, 1974.

[1976] A reducibility condition for recursiveness, *Proc. Am. Math. Soc.* 60 (1976) 270–272.

Mostowski, A.

[1952] *Sentences undecidable in formalized arithmetic*, North Holland, 1952.

Mourad, K.J.

[199?] The Sacks Splitting Theorem and Σ_1-induction, *to appear*.

[199?a] The equivalence of the basis theorem and induction, *to appear*.

Muchnik, A.A.

[1956] Negative answer to the problem of reducibility in the theory of algorithms, *Dokl. Acad. Nauk* 108 (1956) 194–197.

[1958] Isomorphisms of systems of recursively enumerable sets with effective properties, *Trud. Mosk. Math. Obsc.* 7 (1958) 407–412, A.M.S. transl. 28 (1963) 7–13.

[1958a] Solution of Post's reduction problem and of certain other problems in the theory of algorithms, *Trud. Mosk. Math. Obsc.* 7 (1958) 391–401, A.M.S. transl. 29 (1963) 197–215.

[1970] On two approaches to classification of recursive functions, in *Problems of Mathematical Logic*, Mir, 1970, pp. 123–138.

Muchnik, S.S.

[1976] The vectorized Grzegorczyk hierarchy, *Zeit. Math. Log. Math. Log.* 22 (1976) 441–480.

[1976a] Computational complexity of multiple recursion schemata, *S.I.A.M. J. Comp.* 5 (1976) 427–451.

Mueller, W.

[1991] *Abstract degree structures*, Ph.D. Thesis, Mount Holyoke College, South Hadley, 1991.

Muller, H.

[1973] Characterization of the elementary functions in terms of depth of nesting of primitive recursive functions, *Rec. Funct. Th. Newslett.* 5 (1973) 14–15.

Myhill, J.

[1953a] A stumbling block in constructive mathematics, *J. Symb. Log.* 18 (1953) 190–191.

[1956] The lattice of recursively enumerable sets, *J. Symb. Log.* 21 (1956) 220.

[1960] Linear bounded automata, *Univ. Penn. Rep.* 22 (1960).

[1961a] Note on degrees of partial functions, *Proc. Am. Math. Soc.* 12 (1961) 519–521.

Mytilinaios, M.

[1989] Finite injury and Σ_1-induction, *J. Symb. Log.* 54 (1989) 38–49.

Mytilinaios, M., and Slaman, T.A.

[1988] Σ_2-collection and the infinite injury priority method, *J. Symb. Log.* 53 (1988) 212–221.

Nepomniaschki, V.A.

[1966] On a basis for recursively enumerable sets, *Dokl. Acad. Nauk* 170 (1966) 1262–1264, transl. 7 (1966) 1369–1372.

[1966a] On certain automata capable of computing a basis for recursively enumerable sets, *Alg. Log.* 5 (1966) 69–83.

[1970] Rudimentary interpretation of two-tape Turing computations, *Kibern.* 2 (1970) 29–35.

[1970a] Rudimentary predicates and Turing computations, *Dokl. Acad. Nauk* 195 (1970) 282–284, transl. 11 (1970) 1462–1465.

Nerode, A.

[1966] Diophantine correct non-standard models in the isols, *Ann. Math.* 84 (1966) 421–432.

Nerode, A., and Shore, R.A.

[1980] Second-order logic and first-order theories of reducibility orderings, in *Kleene symposium*, Barwise et al. eds., North Holland, 1980, pp. 181–200.

[1980a] Reducibility orderings: theories, definability and automorphisms, *Ann. Math. Log.* 18 (1980) 61–89.

Nies, A.

[1992] *Definability and undecidability in recursion-theoretic semilattices*, Ph.D. Thesis, Heidelberg University, 1992.

[1993] Interpreting true arithmetic in degree structures, *Springer Lect. Not. Comp. Sci.* 713 (1993) 255–263.

[1994] The last question on recursively enumerable many-one degrees, *Alg. Log.* 33 (1994) 550–563.

[1996] Undecidable fragments of elementary theories, *Alg. Univ.* 35 (1996) 8–33.

[1996a] Relativization of structures arising from computability theory, *Lond. Math. Soc. Lect. Not.* 224 (1996) 219–232.

[1997] A uniformity of degree structures, in *Complexity, logic and Recursion Theory*, Sorbi ed., Dekker, 1997, pp. 261–276.

[1997a] Intervals of the lattice of computably enumerable sets and effective Boolean algebras, *Bull. Lond. Math. Soc.* 29 (1997) 683–692.

[1998] *Coding methods in Computability Theory and Complexity Theory*, Habilitationsschrift, Heidelberg University, 1998.

[199?] Interpreting the natural numbers in the computably enumerable weak truth-table degrees, *to appear*.

[199?a] Effectively dense Boolean algebras and their applications, *to appear*.

[199?b] Undecidable fragments of the theory of $\boldsymbol{\mathcal{E}}^*$, *to appear*.

Nies, A., and Shore, R.A.

[1995] Interpreting true arithmetic in the theory of the r.e. truth-table degrees, *Ann. Pure Appl. Log.* 75 (1995) 269–311.

Nies, A., Shore, R.A., and Slaman, T.A.

[1998] Interpretability and definability in the recursively enumerable degrees, *Proc. Lond. Math. Soc.* 77 (1998) 241–291.

Nies, A., and Sorbi, A.

[199?] Structural properties and Σ_2^0 enumeration degrees, *to appear*.

[199?a] Branching in the Σ_2^0 enumeration degrees, *to appear*.

Odifreddi, P.G.

[1976] Note sugli insiemi implicitamente definibili, *Rend. Sem. Mat. Univ. Torino* 34 (1976) 327–332.

[1981] Strong reducibilities, *Bull. Am. Math. Soc.* 4 (1981) 37–86.

[1983] On the first-order theory of the arithmetical degrees, *Proc. Am. Math. Soc.* 87 (1983) 505–507.

[1983a] Forcing and reducibilities, *J. Symb. Log.* 48 (1983) 288–310.

[1989] *Classical Recursion Theory*, vol. I, North Holland, 1989.

[1996] Inductive inference of total functions, *Lond. Math. Soc. Lect. Not.* 224 (1996) 259–288.

[1999] Reducibilities, in Griffor [1999], pp. 89–120.

Odifreddi, P.G., and Shore, R.A.

[1991] Global properties of local structures of degrees, *Boll. Un. Mat. Ital.* 5 (1991) 97–120.

Ogiwara, M., and Watanabe, O.

[1991] On polynomial-time bounded truth-table reducibility of NP to sparse sets, *S.I.A.M. J. Comp.* 20 (1991) 471–483.

Omanadze, R.

[1984] On the uppersemilattice of recursively enumerable Q-degrees, *Alg. Log.* 23 (1984) 175–184, transl. 23 (1984) 124–130.

Orponen, P.

[1986] A classification of complexity core lattices, *Theor. Comp. Sci.* 47 (1986) 121–130.

Orponen, P., and Schöning, U.

[1986] On the density and complexity of polynomial cores for intractable sets, *Inf. Contr.* 70 (1986) 54–68.

Osherson, D.N., Stob, M., and Weinstein, S.

[1982] Learning strategies, *Inf. Contr.* 53 (1982) 32–51.

[1982a] Note on a central lemma for learning theory, *J. Math. Psych.* 27 (1982) 86–92.

[1986] *Systems that learn: an introduction for cognitive and computer scientists*, MIT Press, 1986.

Osherson, D.N., and Weinstein, S.

[1982] Criteria of language learning, *Inf. Contr.* 52 (1982) 123–138.

Owings, J.C.

[1967] Recursion, metarecursion and inclusion, *J. Symb. Log.* 32 (1967) 173–178.

Papadimitriou, C.H..

[1981] On the complexity of integer programming, *J. Ass. Comp. Mach.* 28 (1981) 765–769.

[1985] Games against nature, *J. Comp. Syst. Sci.* 31 (1985) 288–301.

[1994] *Computational Complexity*, Addison Wesley, 1994.

Papadimitriou, C.H., and Steiglitz, K.

[1982] *Combinatorial optimization*, Prentice Hall, 1982.

Parikh, R.

[1967] On nonuniqueness in transfinite progressions, *J. Ind. Math. Soc.* 31 (1967) 23–32.

[1971] Existence and feasibility in arithmetic, *J. Symb. Log.* 36 (1971) 494–508.

[1973] Some results on the length of proofs, *Trans. Am. Math. Soc.* 177 (1973) 29–36.

Park, D.M.

[1970] The $\mathcal{Y}$ combinator in Scott's Lambda Calculus models, *Mimeographed notes*, University of Warwick, 1970.

Parsons, C.

[1968] Hierarchies of primitive recursive functions, *Zeit. Math. Log. Grund. Math.* 14 (1968) 357–376.

[1970] On number-theoretic choice schema and its relation to induction, in *Intuitionism and Proof Theory*, Kino et al. eds., North Holland, 1970, pp. 459–473.

Paterson, M.S.

[1972] Tape bounds for time-bounded Turimg machines, *J. Comp. Syst. Sci.* 6 (1972) 116–124.

Paul, W.J.

[1979] On time hierarchies, *J. Comp. Syst. Sci.* 19 (1979) 197–202.

Paul, W.J., Pippinger, N., Szemerédi, E., and Trotter, W.T.

[1983] On determinism versus nondeterminism and related problems, *Proc. Symp. Found. Comp. Sci.* 24 (1983) 429–438.

Paul, W.J., and Reischuck, R.

[1981] On time versus space II, *J. Comp. Syst. Sci.* 22 (1981) 312–327.

Paul, W.J., Seiferas, J.I., and Simon, J.

[1981] An information-theoretic approach to time bounds for on-line computation, *J. Comp. Syst. Sci.* 23 (1981) 108–126.

Perrin, D.

[1990] Finite automata, in *Handbook of Theoretical Computer Science*, Volume B, van Leeuwen ed., North Holland, 1990, pp. 1–57.

Peter, R.

[1934] Über den Zusammenhang der verschiedenen Begriffe der rekursiven Funktion, *Math. Ann.* 110 (1934) 612–632.

[1935] Konstruktion nichtrekursiver Funktionen, *Math. Ann.* 111 (1935) 42–60.

[1936] Über die mehrfache Rekursion, *Math. Ann.* 113 (1936) 489–527.

[1950] Mehrfache und transfinite Rekursionen, *J. Symb. Log.* 15 (1950) 248–272.

[1951] *Recursive Funktionen*, Akadémiai Kiadó, 1951, transl. Academic Press, 1967.

Pippinger, N.

[1979] On simultaneous resource bounds, *Proc. Symp. Found. Comp. Sci.* 20 (1979) 307–311.

Pitt, L., and Smith, C.H.

[1988] Probability and plurality for aggregations of learning machines, *Inf. Comp.* 77 (1988) 77–92.

Plotkin, G.D.

[1972] A set-theoretical definition of application, *Mimeographed notes*, 1972.

Podnieks, K.M.

[1974] Comparing various concepts of functions prediction, I, *Latv. Gos. Univ. Učen. Zap.* 210 (1974) 68–81.

[1975] Comparing various concepts of functions prediction, II, *Latv. Gos. Univ. Učen. Zap.* 233 (1975) 33–44.

Pohlers, W.

[1992] A short course in ordinal analysis, in *Proof Theory*, Aczel et al. eds., Cambridge University Press, 1992, pp. 27–78.

Popper, K.

[1934] *The logic of scientific discovery*, 1934.

Posner, D.

[1977] *High degrees*, Ph.D. Thesis, University of California, Berkeley, 1977.

[1981] A survey of non r.e. degrees below $\mathbf{0}'$, *Lond. Math. Soc. Lect. Not.* 45 (1981) 52–109.

[1981a] The uppersemilattice of degrees below $\mathbf{0}'$ is complemented, *J. Symb. Log.* 46 (1981) 705–713.

Posner, D., and Robinson, R.W.

[1981] Degrees joining to $\mathbf{0}'$, *J. Symb. Log.* 46 (1981) 714–722.

Post, E.L.

[1944] Recursively enumerable sets of positive integers and their decision problem, *Bull. Am. Math. Soc.* 50 (1944) 284–316.

[1946] Note on a conjecture of Skolem, *J. Symb. Log.* 11 (1946) 73–74.

Putnam, H.

[1963a] Probability and confirmation, in Putnam [1975].

[1975] *Mathematics, matter and method*, Volume I, Cambridge University Press, 1975.

Rabin, M.O.

[1960a] Degree of difficulty of computing a function and a partial ordering of the recursive sets, *Univ. Jerus. Rep.* 2 (1960).

[1963] Real-time computations, *Isr. J. Math.* 1 (1963) 203–211.

[1969] Decidability of second order theories and automata on infinite trees, *Trans. Am. Math. Soc.* 141 (1969) 1–35.

Rabin, M.O., and Scott, D.

[1959] Finite automata and their decision precision, *I.B.M. J. Res.* 3 (1959) 115–125.

Rackoff, C.

[1982] Relativized questions involving probabilistic algorithms, *J. Ass. Comp. Mach.* 29 (1982) 261–268.

Regan, K.W.

[1983] On diagonalization methods and the structure of language classes, *Springer Lect. Not. Comp. Sci.* 158 (1983) 368–380.

[1992] Diagonalization, uniformity, and fixed-point theorems, *Inf. Comp.* 98 (1992) 1–40.

Rice, H.G.

[1953] Classes of recursively enumerable sets and their decision problems, *Trans. Am. Math. Soc* 74 (1953) 358–366.

Ritchie, D.M.

[1965] Complexity classification of primitive recursive functions by their machine programs, *Not. Am. Math. Soc.* 12 (1965) 343.

Ritchie, R.W.

[1963] Classes of predictably computable functions, *Trans. Am. Math. Soc.* 106 (1963) 139–173.

[1965] A rudimentary definition of addition, *J. Symb. Log.* 30 (1965) 350–354.

[1965a] Classes of recursive functions based on Ackermann's function, *Pac. J. Math.* 15 (1965) 1027–1044.

Robbin, J.W.

[1965] *Subrecursive hierarchies*, Ph.D. Thesis, Princeton University, 1965.

Robertson, E.L.

[1974] Complexity classes of partial recursive functions, *J. Comp. Syst. Sci.* 9 (1974) 69–87.

Robinson, R.M.

[1947] Primitive recursive functions, *Bull. Am. Math. Soc.* 53 (1947) 925–942.

[1948] Recursion and double recursion, *Bull. Am. Math. Soc.* 54 (1948) 987–993.
[1955] Primitive recursive functions, II, *Proc. Am. Math. Soc.* 6 (1955) 663–666.

Robinson, R.W.
[1966] *The inclusion lattice and degrees of unsolvability of the recursively enumerable degrees*, Ph.D. Thesis, Cornell University, 1966.
[1967] Two theorems on hyperhypersimple sets, *Trans. Am. Math. Soc.* 128 (1967) 531–538.
[1967a] Simplicity of recursively enumerable sets, *J. Symb. Log.* 32 (1967) 162–172.
[1968] A dichotomy of the recursively enumerable sets, *Zeit. Math. Log. Grund. Math.* 14 (1968) 339–356.
[1969] Review of Sacks [1964a], *J. Symb. Log.* 34 (1969) 294–295.
[1971] Interpolation and embeddings in the recursively enumerable degrees, *Ann. Math.* 93 (1971) 285–314.
[1971a] Jump restricted interpolation in the r.e. degrees, *Ann. Math.* 93 (1971) 586–596.

Rödding, D.
[1964] Über die Elimnierebarkeit von Definitionskemata in der Theorie der rekursiven Funktionen, *Zeit. Math. Log. Grund. Math.* 10 (1964) 315–330.
[1965] Darstellungen der (in Kalmar-Csillag schen Sinne) elementaren Funktionen, *Arch. Math. Log.* 7 (1965) 139–158.
[1966] Über Darstellungen der elementaren Funktionen, II, *Arch. Math. Log.* 9 (1966) 36–48.

Rogers, H.
[1959] Computing degrees of unsolvability, *Math. Ann.* 138 (1959) 125–140.
[1967] *Theory of recursive functions and effective computability*, McGraw Hill, 1967.

Rose, H.E.
[1984] *Subrecursion*, Clarendon Press, 1984.

Rosenberg, A.
[1967] Real-time definable languages, *J. Ass. Comp. Mach.* 14 (1967) 645–662.

Rosser, B.J.
[1936] Extensions of some theorems of Gödel and Church, *J. Symb. Log.* 1 (1936) 87–91, also in Davis [1965], pp. 230–235.

Routledge, N.A.
[1953] Ordinal recursion, *Proc. Cambr. Phil. Soc.* 49 (1953) 175–182.

Rozinas, M.G.
[1978] The semilattice of e-degrees, in *Recursive functions*, Ivanovo State University, 1978, pp. 71–84.

Rubin, M.
[1976] The theory of Boolean algebras with distinguished subalgebras is undecidable, *Ann. Sci. Univ. Clermont Math.* 13 (1976) 129–134.

Rustin, R., ed.
[1973] *Computational complexity*, Algorithmic Press, 1973.

Ruzzo, W.L., Simon, J., and Tompa, M.
[1984] Space-bounded hierarchies and probabilistic computations, *J. Comp. Syst. Sci.* 28 (1984) 216–230.

Sacks, G.E.
[1961a] A minimal degree below $\mathbf{0}'$, *Bull. Am. Math. Soc.* 67 (1961) 416–419.
[1963] *Degrees of unsolvability*, Princeton University Press, 1963, 2nd ed. 1966.
[1963a] On the degrees less than $\mathbf{0}'$, *Ann. Math.* 77 (1963) 211–231.

[1963b] Recursive enumerability and the jump operator, *Trans. Am. Math. Soc.* 108 (1963) 223–239.
[1964a] The recursively enumerable degrees are dense, *Ann. Math.* 80 (1964) 300–312.
[1964b] A maximal set which is not complete, *Mich. Math. J.* 11 (1964) 193–205.
[1966] On a theorem of Martin and Lachlan, *Proc. Am. Math. Soc.* 17 (1966) 140–141.
[1966a] Post's Problem, admissible ordinals and regularity, *Trans. Am. Math. Soc.* 124 (1966) 1–23.
[1969] Measure-theoretic uniformity in recursion theory and set theory, *Trans. Am. Math. Soc.* 142 (1969) 381–420.
[1971] Forcing with perfect closed sets, *Proc. Symp. Pure Appl. Math.* 17 (1971) 331–355.
[1990] *Higher Recursion Theory*, Springer Verlag, 1990.

Sacks, G.E., and Simpson, S.G.
[1972] The α-finite injury method, *Ann. Math. Log.* 4 (1972) 323–367.

Salomaa, A.
[1973] *Formal languages*, Academic Press, 1973.

Sasso, L.P.
[1970] A cornucopia of minimal degrees, *J. Symb. Log.* 35 (1970) 382–388.
[1974] A minimal degree not realizing the least possible jump, *J. Symb. Log.* 39 (1974) 571–573.
[1974a] Deficiency sets and bounded information reducibilities, *Trans. Am. Math. Soc.* 200 (1974) 267–290.

Savage, J.E.
[1976] *The complexity of computing*, Wiley, 1976.

Savitch, W.J.
[1970] Relationships between non deterministic and deterministic tape complexity, *J. Comp. Syst. Sci.* 4 (1970) 177–192.
[1983] A note on relativized log-space, *Math. Syst. Th.* 16 (1983) 229–235.

Schaefer, M.
[1998] A guided tour of minimal indices and shortest descriptions, *Arch. Math. Log.* 37 (1998) 521–548.

Schmerl, U.R.
[1982] Number theory and the Bachmann-Howard ordinal, in *Logic Colloqium '81*, Stern ed., North Holland, 1982, pp. 287–298.

Schmidt, D.
[1976] Built-up systems of fundamental sequences and hierarchies of number-theoretic functions, *Arch. Math. Log.* 18 (1976) 47–53.
[1977] Postscript to [1976], *Arch. Math. Log.* 18 (1977) 145–146.
[1984] On the complement of one complexity class in another, *Springer Lect. Not. Comp. Sci.* 171 (1984) 77–87.
[1985] The recursion-theoretic structure of complexity classes, *Theor. Comp. Sci.* 38 (1985) 143–156.

Schnorr, C.P.
[1973] Does the computational speed-up concern programming?, in *Automata, languages and programming*, Nivat ed., North Holland, 1973, pp. 585–591.

Schnorr, C.P., and Stumpf, G.
[1975] A characterization of complexity sequences, *Zeit. Math. Log. Grund. Math.* 21 (1975) 47–56.

Schöning, U.
[1982] A uniform approach to obtain diagonal sets in complexity classes, *Theor. Comp.*

Sci. 18 (1982) 95–103.
[1983] A low and high hierarchy within NP, *J. Comp. Syst. Sci.* 27 (1983) 14–28.
[1984] Minimal pairs for P, *Theor. Comp. Sci.* 31 (1984) 41–48.
[1986] *Complexity and structure*, Springer Verlag, 1986.
[1990] The power of counting, in *Complexity Theory Retrospective*, Selman ed., Springer, 1990, pp. 204–223.

Schöning, U, and Book, R.
[1984] Immunity, relativizations, and nondeterminism, *S.I.A.M. J. Comp.* 13 (1984) 329–337.

Schöning, U, and Wagner, K.
[1988] Collapsing hierarchies, census functions, and logarithmically many queries, *Springer Lect. Not. Comp. Sci.* 294 (1988) 91–98.

Schütte, K.
[1977] *Proof Theory*, Springer, 1977.

Schwarz, S.T.
[1984] The quotient semilattice of the recursively enumerable degrees modulo the capable degrees, *Trans. Am. Math. Soc.* 283 (1984) 315–328.
[1989] Index sets related to prompt simplicity, *Ann. Pure Appl. Log.* 42 (1989) 243–254.
[199?] Index sets related to the high-low hierarchy, *to appear.*

Schwichtenberg, H.
[1969] Rekursionzahlen und die Grzegorczyk-Hierarchie, *Arch. Math. Log.* 12 (1969) 85–97.
[1971] Eine Klassifikation der ϵ_0-rekursiven Funktionen, *Zeit. Math. Log. Grund. Math.* 17 (1971) 61–74.
[1972] Beweistheoretische Charakterisierung einer Erweiterung der Grzegorczyk-Hierarchie, *Arch. Math. Log.* 15 (1972) 129–145.
[1999] Classifying recursive functions, in Griffor [1999], pp. 705–757.

Scott, D.
[1975] Lambda Calculus and Recursion Theory, in *Proceedings of the Third Scandinavian Logic Symposium*, Kanger ed., North Holland, 1975, pp. 154–193.

Seetapun, D., and Slaman, T.A.
[199?] Minimal complements, *to appear*.

Seiferas, J.I.
[1977] Techniques for separating space complexity classes, *J. Comp. Syst. Sci.* 14 (1977) 73–99.
[1977a] Relating refined space complexity classes, *J. Comp. Syst. Sci.* 14 (1977) 100–129.

Seiferas, J.I., Fischer, M.J., and Meyer, A.R.
[1978] Separating non deterministic time complexity classes, *J. Ass. Comp. Mach.* 25 (1978) 146–167.

Selivanov, V.L.
[1978] On the index sets of computable classes of finite sets, *Alg. Autom.* 10 (1978) 95–99.
[1978a] Some remarks on classes of recursively enumerable sets, *Sib. Math. J.* 19 (1978) 153–160, transl. 19 (1978) 109–113.
[1979] On the structure of degrees of unsolvability of index sets, *Alg. Log.* 18 (1979) 463–480, transl. 18 (1979) 286–299.
[1982] On one class of reducibilities in the theory of recursive functions, *Probl. Math. Cyb.* 18 (1982) 83–100.

[1982a] On index sets in the Kleene-Mostowski hierarchy, *Math. Log. Th. Alg.* 2 (1982) 135–158.
[1984] On a hierarchy of limiting computations, *Sib. Math. J.* 25 (1984) 146–156.
[1984a] Index sets in the hyperarithmetical hierarchy, *Sib. Math. J.* 25 (1984) 164–181, transl. 25 (1984) 474–488.
[1985] On Ershov's hierarchy, *Sib. Math. J.* 26 (1985) 134–149.

Selman, A.L.
[1971] Arithmetical reducibilities I, *Zeit. Math. Log. Grund. Math.* 17 (1971) 335–370.
[1972] Applications of forcing to the degree theory of the arithmetical hierarchy, *Proc. Lond. Math. Soc.* 25 (1972) 586–602.
[1974] Relativized halting problem, *Zeit. Math. Log. Grund. Math.* 20 (1974) 193–198.
[1978] Polynomial time enumeration reducibility, *S.I.A.M. J. Comp.* 7 (1978) 440–457.
[1979] P-selective sets, tally languages, and the behaviour of polynomial time reducibilities on NP, *Math. Syst. Th.* 13 (1979) 55–65.
[1982] Reductions on NP and P-selective sets, *Theor. Comp. Sci.* 19 (1982) 287–304.
[1982a] Analogues of semirecursive sets and effective reducibilities to the study of NP complexity, *Inf. Contr.* 52 (1982) 36–51.
[1992] A survey of one-way functions in Complexity Theory, *Math. Syst. Th.* 25 (1992) 203–221.

Selman, A.L., Xu, M., and Book, R.
[1983] Positive relativizations of complexity classes, *S.I.A.M. J. Comp.* 12 (1983) 656–579.

Sewelson, V.D.
[1983] *A study of the structure of* NP, Ph.D. Thesis, Cornell University, 1983.

Shamir, A.
[1990] IP = PSPACE, *Proc. Symp. Found. Comp. Sci.* 31 (1990) 11–15.

Shapiro, N.
[1956] Degrees of computability, *Trans. Am. Math. Soc.* 82 (1956) 281–299.

Shepherdson, J.C.
[1959] The reduction of two-way automata to one-way automata, *I.B.M. J. Res.* 3 (1959) 198–200.

Shinoda, J., and Slaman, T.A.
[1990] On the theory of PTIME degrees of the recursive sets, *J. Comp. Syst. Sci.* 41 (1990) 321–366.

Shoenfield, J.R.
[1958a] The class of recursive functions, *Proc. Am. Math. Soc.* 9 (1958) 690–692.
[1959] On degrees of unsolvability, *Ann. Math.* 69 (1959) 644–653.
[1961] Undecidable and creative theories, *Fund. Math.* 49 (1961) 171–179.
[1966] A theorem on minimal degrees, *J. Symb. Log.* 31 (1966) 539–544.
[1971] Unramified forcing, *Proc. Symp. Pure Appl. Math.* 13 (1971) 357–381.
[1971a] *Degrees of unsolvability*, North Holland, 1971.
[1976] Degrees of classes of recursively enumerable sets, *J. Symb. Log.* 41 (1976) 695–696.
[1990] Non-bounding constructions, *Ann. Pure Appl. Log.* 50 (1990) 191–205.

Shore, R.A.
[1975] Splitting an α-recursively enumerable set, *Trans. Am. Math. Soc.* 204 (1975) 65–78.
[1975a] The irregular and non-hyperregular α-r.e. degrees, *Isr. J. Math.* 22 (1975) 28–41.
[1976] The recursively enumerable α-degrees are dense, *Ann. Math. Log.* 9 (1976) 123–155.

[1976a] On the jump of the recursively enumerable α-degrees, *Trans. Am. Math. Soc.* 217 (1976) 351–363.

[1977] Determining automorphisms of the recursively enumerable sets, *Proc. Am. Math. Soc.* 65 (1977) 318–325.

[1977a] α-recursion theory, in *Handbook of Mathematical Logic*, Barwise ed., North Holland, 1977, pp. 653–680.

[1978] On the $\forall\exists$ sentences of α-recursion theory, in *Generalized recursion theory*, Fenstad et al. eds., North Holland, 1978, pp. 331–353.

[1978a] Nowhere simple sets and the lattice of recursively enumerable sets, *J. Symb. Log.* 43 (1978) 322–330.

[1981] The degrees of unsolvability: global results, *Springer Lect. Not. Math.* 859 (1981) 283–301.

[1981a] The theory of degrees below $\mathbf{0}'$, *J. Lond. Math. Soc.* 24 (1981) 1–14.

[1982b] Finitely generated coding and the degrees r.e. in a degree $\boldsymbol{d}$, *Proc. Am. Math. Soc.* 84 (1982) 256–263.

[1984] The arithmetic and Turing degrees are not elementarily equivalent, *Arch. Math. Log. Grund.* 24 (1984) 137–139.

[1988] Defining jump classes in the degrees below $\mathbf{0}'$, *Proc. Am. Math. Soc.* 104 (1988) 287–292.

[1988a] A non-inversion theorem for the jump operator, *Ann. Pure Appl. Log.* 40 (1988) 277–303.

[1993] The theories of the T, *tt* and *wtt* r.e. degrees: undecidability and beyond, *Notas de Logica Matematica* 38 (1993) 61–70.

[1997] Conjectures and questions from Gerald Sacks' *Degrees of unsolvability*, *Arch. Math. Log.* 36 (1997) 233–253.

[1999] The recursively enumerable degrees, in Griffor [1999], pp. 169–198.

Shore, R.A., and Slaman, T.A.

[1990] Working below a low$_2$ recursively enumerable degree, *Arch. Math. Log.* 29 (1990) 201–211.

[1992] The p-T-degrees of the recursive sets: lattice embeddings, extensions of embeddings and the two-quantifier theory, *Theor. Comp. Sci.* 97 (1992) 263–284.

[1993] Working below a high recursively enumerable degree, *J. Symb. Log.* 58 (1993) 824–859.

Simon, I.

[1977] Polynomially bounded quantification over higher types and a new hierarchy of the elementary sets, in *Non-Classical Logic, Model Theory and Computability*, Arruda et al. eds., North Holland, 1977, 267–281.

Simpson, M.F.

[1985] *Arithmetic degrees: initial segments, ω-r.e.a. operators and the ω-jump*, Ph.D. Thesis, Cornell University, 1985.

Simpson, S.G.

[1974] Post's Problem for admissible sets, in *Generalized Recursion Theory*, Fenstad et al. eds., North Holland, 1974, pp. 437–441.

[1987] Unprovable theorems and fast-growing functions, in *Logic and Combinatorics*, American Mathematical Society, 1987, pp. 359–394.

Sipser, M.

[1980] Halting space-bounded computations, *Theor. Comp. Sci.* 10 (1980) 335–338.

[1982] On relativization and the existence of complete sets, *Springer Lect. Not. Comp. Sci.* 140 (1982) 523–531.

[1983] A complexity-theoretic approach to randomness, *Proc. Symp. Th. Comp.* 15 (1983) 330–335.

[1992] The history and status of the P versus NP problem, *Proc. Symp. Th. Comp.* 24 (1992) 603–618.

Skolem, T.

[1923] Begründung der elementaren Arithmetik durch die rekurriende Denkweise ohne Anwendung scheinbarer Veränderlichen mit unendlichem Ausdehnungsbereich, *Vidensk. Skrifter*, no. 6, transl. in Van Heijenoort [1967], pp. 303–333.

[1934] Über die Nicht-charakterisierbarkeit der Zahlenreihe mittels endlich oder abzählbar unendlich vieler Aussagen mit ausschliesslich Zahlenvaiablen, *Fund. Math.* 23 (1934) 150–161.

[1944] Remarks on recursive functions and relations, *Kon. Nor. Vid. Sel. Forh.* 17 (1944) 89–92.

[1944a] Some remarks on recursive arithmetic, *Kon. Nor. Vid. Sel. Forh.* 17 (1944) 103–106.

[1954] Remarks on "elementary" arithmetical functions, *Kon. Nor. Vid. Sel. Forh.* 27 (1954) 1–6.

[1954a] Some considerations concerning recursive arithmetic, *Bull. Soc. Math. Belg.* 6 (1954) 35–46.

[1962] Proofs of some theorems on recursively enumerable sets, *Notre Dame J. Form. Log.* 3 (1962) 65–74.

[1963] Addendum to [1962], *Notre Dame J. Form. Log.* 4 (1963) 44–47.

Slaman, T.A.

[1991] The density of infima in the recursively enumerable degrees, *Ann. Pure Appl. Log.* 52 (1991) 1–25.

[1991a] Degree structures, in *Proceedings of the 1990 International Congress of Mathematics*, Springer, 1991, pp. 303–316.

[1999] The global structure of Turing degrees, in Griffor [1999], pp. 155–168.

[199?] The recursively enumerable degrees as a substructure of the Δ_2^0 degrees, *to appear*.

[199?a] A recursively enumerable degree that is not the top of a diamond in the Turing degrees, *to appear*.

Slaman, T.A., and Soare, R.I.

[1995] Algebraic aspects of the computably enumerable degrees, *Proc. Nat. Acad. Sci.* 92 (1995) 617–621.

[199?] Extensions of embeddings in the recursively enumerable degrees, *to appear*.

Slaman, T.A., and Solovay, R.M.

[1991] When oracles do not help, *Proc. Conf. Comp. Learn. Th.* 4 (1991) 379–383.

Slaman, T.A., and Steel, J.

[1989] Complementation in the Turing degrees, *J. Symb. Log.* 54 (1989) 160–176.

Slaman, T.A., and Woodin, H.W.

[1986] Definability in the Turing degrees, *Ill. J. Math.* 30 (1986) 320–334.

[1989] Σ_1-collection and the finite injury priority method, *Springer Lect. Not. Math.* 1388 (1989) 178–188.

[1997] Definability in the enumeration degrees, *Arch. Math. Log.* 36 (1997) 255–267.

[199?] *Definability in degree structures*, *to appear*.

Smale, S.

[1983] On the average number of steps of the simplex method of linear programming, *Math. Prog.* 27 (1983) 241–263.

Smith, C.H.

[1982] The power of pluralism for automatic program synthesis, *J. Ass. Comp. Mach.* 29 (1982) 1144–1165.

[1988] A note on arbitrarily complex recursive functions, *Notre Dame J. Form. Log.* 29 (1988) 198–207.

Smorynski, C.

[1980] Some rapidly growing functions, *Math. Intell.* 2 (1980) 149–154.

[1982] The varieties of arboreal experience, *Math. Intell.* 4 (1982) 182–189.

[1983] Big news from Archimedes to Friedman, *Not. Am. Math. Soc.* 30 (1983) 251–256.

Smullyan, R.M.

[1961] *Theory of formal systems*, Princeton University Press, 1961.

Soare, R.I.

[1969] Recursion theory and Dedekind cuts, *Trans. Am. Math. Soc.* 140 (1969) 271–294.

[1972] The Friedberg-Muchnik theorem re-examined, *Can. J. Math.* 24 (1972) 1070–1078.

[1974] Automorphisms of the lattice of recursively enumerable sets, *Bull. Am. Math. Soc.* 80 (1974) 53–58.

[1974a] Automorphisms of the lattice of recursively enumerable sets I: maximal sets, *Ann. Math.* 100 (1974) 80–120.

[1976] The infinite injury priority method, *J. Symb. Log.* 41 (1976) 513–530.

[1977] Computational complexity, speedable and levelable sets, *J. Symb. Log.* 42 (1977) 545–563.

[1982] Automorphisms of the lattice of recursively enumerable sets II: low sets, *Ann. Math. Log.* 22 (1982) 69–108.

[1985] Tree arguments in Recursion Theory and the $\emptyset'''$-priority method, *Proc. Symp. Pure Math.* 42 (1985) 53–106.

[1987] *Recursively enumerable sets and degrees*, Springer, 1987.

[1999] An overview of the computably enumerable sets, in Griffor [1999], pp. 199-249.

Solomonoff, R.

[1964] A formal theory of inductive inference, I, *Inf. Contr.* 7 (1964) 1–22.

[1964a] A formal theory of inductive inference, II, *Inf. Contr.* 7 (1964) 224–254.

Solovay, R.M.

[1976] On sets Cook-reducible to sparse sets, *S.I.A.M. J. Comp.* 5 (1976) 646–652.

Soloviev, V.D.

[1974] Q-reducibility and hyperhypersimple sets, *Probl. Meth. Cyb.* 10 (1974) 121–128.

Sorbi, A.

[1997] The enumeration degrees of Σ^0_2 sets, in *Complexity, logic and Recursion Theory*, Sorbi ed., Dekker, 1997, pp. 303–330.

[1998] Sets of generators and automorphism bases for the enumeration degrees, *Ann. Pure Appl. Log.* 94 (1998) 263–272.

Spector, C.

[1956] On degrees of recursive unsolvability, *Ann. Math.* 64 (1956) 581–592.

Statman, R.

[1978] Bounds for proof-search and speed-up in the predicate calculus, *Ann. Math. Log.* 15 (1978) 225–287.

[1981] Speed-up by theories with infinite models, *Proc. Am. Math. Soc.* 81 (1981) 465–469.

[1979] Intuitionistic Propositional Calculus is PSPACE-complete, *Theor. Comp. Sci.* 9 (1979) 67–72.

Steel, J.R.

[1982] A classification of jump operators, *J. Symb. Log.* 47 (1982) 347–358.

Stephan, F.

[199?] On the structure inside truth-table degrees, *to appear*.

Stob, M.

[1979] *The structure and elementary theory of the recursively enumerable degrees*, Ph.D. Thesis, University of Chicago, 1979.

[1982] Index sets and degrees of unsolvability, *J. Symb. Log.* 47 (1982) 241–248.

[1982a] Invariance of properties under automorphisms of the lattice of recursively enumerable sets, *Pac. J. Math.* 100 (1982) 445–471.

[1983] *wtt*-degrees and T-degrees of recursively enumerable sets, *J. Symb. Log.* 48 (1983) 921–930.

[1985] Major subsets and the lattice of recursively enumerable sets, *Proc. Symp. Pure Appl. Math.* 42 (1985) 107–116.

Stockmeyer, L.J.

[1974] *The complexity of decision problems in automata theory and logic*, Ph.D. Thesis, M.I.T., 1974.

[1976] The Polynomial Time Hierarchy, *Theor. Comp. Sci.* 3 (1976) 1–22.

[1985] On approximation algorithms for $\sharp$P, *S.I.A.M. J. Comp.* 14 (1985) 849–861.

Stockmeyer, L.J., and Meyer, A.R.

[1973] Word problems requiring exponential time, *Proc. Symp. Th. Comp.* 5 (1973) 1–9.

Strnad, P.

[1968] On-line Turing machine recognition, *Inf. Contr.* 12 (1968) 442–452.

Sudan, G.

[1927] Sur le nombre transfini ω^ω, *Bull. Soc. Roum. Sci.* 30 (1927) 11-30.

Sui, Y.

[1987] Some results about the r.e. degrees, *Acta Math. Sinica* 3 (1987) 170–179.

[1994] More on minimal pairs, *Acta Math. Sinica* 10 (1994) 220–224.

Sui, Y., and Zhang, Z.

[199?] The Cupping Theorem in $\boldsymbol{\mathcal{R}/M}$, *to appear*.

Szelepcsényi, R.

[1988] The method of forced enumeration for nondeterministic automata, *Acta Inf.* 26 (1988) 279–284.

Tait, W.W.

[1959] A characterization of ordinal recursive functions, *J. Symb. Log.* 24 (1959) 325.

[1961] Nested recursion, *Math. Ann.* 143 (1961) 236–250.

Takeuti, G.

[1975] *Proof Theory*, North Holland, 1975, 2nd ed. 1980.

Tanaka, H.

[1967] Some results in the effective descriptive set theory, *Publ. Res. Inst. Math. Sci. Kyoto Univ.* 3 (1967) 11–52.

[1970] On a Π^0_1 set of positive measure, *Nagoya Math. J.* 38 (1970) 139–144.

[1972] A property of arithmetic sets, *Proc. Am. Math. Soc.* 31 (1972) 521–524.

Tarski, A.

[1936] Der Wahrheitsbegriff in der formalisierten Sprachen, *Studia Phil.* 1 (1936) 261–405, transl. in [1956], pp. 152–278.

[1956] *Logic, semantics, metamathematics*, Oxford University Press, 1956.

Tennenbaum, S.
[1961] Degrees of unsolvability and the rate of growth of functions, *Not. Am. Math. Soc.* 8 (1961) 608.

Thomason, S.K.
[1971] Sublattices of the recursively enumerable degrees, *Zeit. Math. Log. Grund. Math.* 17 (1971) 273–280.

Thompson, D.B.
[1972] Subrecursiveness: machine-independent notions of computability in restricted time and storage, *Math. Syst. Th.* 6 (1972) 3–15.

Thue, A.
[1914] Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln, *Skrif. Viden. Krist.* 10 (1914).

Toda, S.
[1987] Σ_2-SPACE is closed under complement, *J. Comp. Syst. Sci.* 35 (1987) 145–152.
[1989] On the computational power of PP and $\oplus$P, *Proc. Symp. Found. Comp. Sci.* 30 (1989) 514–519.
[1991] PP is as hard as the Polynomial Time Hierarchy, *S.I.A.M. J. Comp.* 20 (1991) 865–877.

Tompa, M.
[1981] An extension of Savitch's theorem to small space bounds, *Inf. Proc. Lett.* 12 (1981) 106–108.

Torenvliet, L., and van Emde Boas, P.
[1989] Simplicity, immunity, relativizations, and nondeterminism, *Inf. Comp.* 80 (1989) 1–17.

Trakhtenbrot, B.A.
[1956] Signalizing functions and tabular operators, *Trans. Penza Ped. Inst.* 4 (1956) 75–87.
[1964] Turing computations with logarithmic delay, *Alg. Log.* 3 (1964) 33–48.
[1965] Optimal computations and the frequency phenomenon of Jablonskii, *Alg. Log.* 4 (1965) 79–93.
[1967] *Complexity of algorithms and computations*, Lecture Notes, Novosibirk, 1967.
[1984] A survey of Russian approaches to perebor (brute-force search) algorithms, *Ann. Hist. Comp.* 6 (1984) 384–400.

Tsichritzis, D.
[1970] Equivalence problems of single programs, *J. Ass. Comp. Mach.* 17 (1970) 729–738.
[1970a] A note on comparison of subrecursive hierarchies, *Inf. Proc. Lett.* 1 (1971) 42–44.

Turing, A.M.
[1936] On computable numbers with an application to the Entscheidungsproblem, *Proc. London Math. Soc.* 42 (1936) 230–265, corrections ibid. 43 (1937) 544–546, also in Davis [1965], pp. 116–154.
[1939] Systems of logic based on ordinals, *Proc. Lond. Math. Soc.* 45 (1939) 161–228, also in Davis [1965], pp. 155–222.

Ukkonen, E.
[1983] Two results on polynomial time truth-table reductions to sparse sets, *S.I.A.M. J. Comp.* 12 (1983) 580–587.

Ullian, J.S.
[1961] A theorem on maximal sets, *Notre Dame J. Form. Log.* 2 (1961) 222–223.

Valiant, L.G.

[1976] Relative complexity of checking and evaluating, *Inf. Proc. Lett.* 5 (1976) 20–23.

Van Emde Boas, P.

[1975] Ten years of speed-up, *Springer Lect. Not. Comp. Sci.* 32 (1975) 13–29.

[1978] Some applications of the McCreight-Meyer algorithm in abstract complexity theory, *Theor. Comp. Sci.* 7 (1978) 79–98.

[1990] Machine models and simulations, in *Handbook of Theoretical Computer Science,* Volume A, van Leeuwen ed., North Holland, 1990, pp. 1–66.

Van Heijenoort, J., ed.

[1967] *From Frege to Gödel*, Harvard University Press, 1967.

Vardi, M.

[1982] Complexity of relational query languages, *Proc. Symp. Th. Comp.* 14 (1982) 137–146.

Veblen, O.

[1908] Continuous increasing functions of finite and transfinite ordinals, *Trans. Am. Math. Soc.* 9 (1908) 280–292.

Volger, H.

[1984] Rudimentary relations and Turing machines with linear alternations, *Springer Lect. Not. Comp. Sci.* 171 (1984) 131–136.

Von Neumann, J.

[1953] A certain zero-sum two-person game equivalent to the optimal assignment problem, *Ann. Math. Stud.* 28 (1953) 5–12.

Wagner, K., and Wechsung, G.

[1986] *Computational complexity*, Reidel, 1986.

Wainer, S.S.

[1970] A classification of the ordinal recursive functions, *Arch. Math. Log.* 13 (1970) 136–153.

[1972] Ordinal recursion, and a refinement of the extended Grzegorczyk hierarchy, *J. Symb. Log.* 37 (1972) 281–292.

[1985] The slow-growing Π^1_2 approach to hierarchies, *Proc. Symp. Pure Math.* 42 (1985) 487–502.

[1989] Slow growing versus fast growing, *J. Symb. Log.* 54 (1989) 608–614.

Wang, J.

[1989] P-creative sets vs. P-completely creative sets, *Proc. Symp. Struct. Compl.* 4 (1989) 24–35.

Watanabe, O.

[1987] A comparison of polynomial time completeness notions, *Theor. Comp. Sci.* 54 (1987) 249–265.

Weiermann, A.

[1995] Investigations on slow versus fast growing: how to majorize slow growing functions nontrivially by fast growing ones, *Arch. Math. Log.* 34 (1995) 313–330.

Werner, G.

[1971] Propriété d'invariance des classes des fonctions de complexité bornée, *Compt. Rend. Acad. Sci.* 273 (1971) 133–136.

[1975] Prognose von Folgen, *J. Inf. Proc. Cyb.* 11 (1975) 649–653.

Wiehagen, R.

[1976] Limes-Erkennung rekursiver Funktionen durch spezielle Strategien, *J. Inf. Proc. Cyb.* 12 (1976) 93–99.

[1977] Identification of formal languages, *Springer Lect. Not. Comp. Sci.* 53 (1977) 571–579.

[1978] Characterization problems in the theory of inductive inference, *Springer Lect. Not. Comp. Sci.* 62 (1978) 494–508.

[1991] A thesis in inductive inference, *Springer Lect. Not. Comp. Sci.* 543 (1991) 184–207.

Wiehagen, R., and Jung, H.

[1977] Rekursionstheoretische Charakterisierung von erkennbaren Klassen rekursiver Funktionen, *J. Inf. Proc. Cyb.* 13 (1977) 385–397.

Wiehagen, R., and Liepe, W.

[1976] Charakteristche Eigenschaften von erkennbaren Klassen rekursiver Funktionen, *J. Inf. Proc. Cyb.* 12 (1976) 421–438.

Wilson, C.

[1985] Relativized circuit complexity, *J. Comp. Syst. Sci.* 31 (1985) 169–181.

Wittgenstein, L.

[1953] *Philosophische Untersuchungen*, Blackwell, 1953.

Wrathall, C.

[1976] Complete sets and the Polynomial Time Hierarchy, *Theor. Comp. Sci.* 3 (1976) 23–33.

Xu, M., Doner, J., and Book, R.

[1983] Refining nondeterminism in controlled relativizations of complexity classes, *J. Ass. Comp. Mach.* 30 (1983) 677–685.

Yamada, H.

[1962] Real-time computation and recursive functions not real-time computable, *Trans. Electr. Comp.* 11 (1962) 753–760.

Yang, D.

[1979] The α-operator gap theorem, *Chin. J. Comp.* 2 (1979) 163–173.

[1987] The existence of minimal honest polynomial degrees below $\mathbf{0}'$, *Chin. Quart. J. Math.* 2 (1987) 21–26.

Yang, Y.

[1995] The Thickness Lemma from $P^- + I\Sigma_1 + \neg B\Sigma_2$, *J. Symb. Log.* 60 (1995) 505–511.

[1995a] Iterated trees and fragments of arithmetic, *Arch. Math. Log.* 34 (1995) 97–112.

Yao, A.C.C.

[1985] Separating the Polynomial Time Hierarchy by oracles, *Proc. Symp. Found. Comp. Sci.* 26 (1985) 1–10.

Yap, C.K.

[1983] Some consequences of nonuniform conditions on uniform classes, *Theor. Comp. Sci.* 26 (1983) 287–300.

Yates, C.E.M.

[1962] Recursively enumerable sets and retracing functions, *Zeit. Math. Log. Grund. Math.* 8 (1962) 331–345.

[1965] Three theorems on the degrees of r.e. sets, *Duke Math. J.* 32 (1965) 461–468.

[1966] A minimal pair of recursively enumerable degrees, *J. Symb. Log.* 31 (1966) 159–168.

[1966a] On the degrees of index sets, *Trans. Am. Math. Soc.* 121 (1966) 309–328.

[1967] Recursively enumerable degrees and the degrees less than $\mathbf{0}'$, in *Models and Recursion Theory*, Crossley et al. eds., North Holland, 1967, pp. 264–271.

[1969] On the degrees of index sets II, *Trans. Am. Math. Soc.* 135 (1969) 249–266.

[1970] Initial segments of the degrees of unsolvability, part I, in *Mathematical logic and* *foundations of set theory*, Bar Hillel ed., North Holland, 1970, pp. 63–83.

[1970a] Initial segments of the degrees if unsolvability, part II: minimal degrees, *J. Symb. Log.* 35 (1970) 243–266.

[1974] A general framework for simple Δ^0_2 and Σ^0_1 priority arguments, *Proc. Int. Congr. Math.* 18 (1974) 269–273.

Yesha, Y.

[1983] On certain polynomial time truth-table reducibilities and sets in NP, *Proc. Symp. Th. Comp.* 15 (1983) 392–401.

Yi, X.

[1996] Extension of embeddings on the recursively enumerable degrees modulo the capable degrees, *Lond. Math. Soc. Lect. Not.* 224 (1996) 313–331.

[199?] Highness and the density property in the *d*-r.e. degrees, *to appear*.

[199?a] A non-splitting theorem for *d*-r.e. sets, *to appear*.

Young, P.R.

[1963] Notes on the structure of recursively enumerable sets, *Not. Am. Math. Soc.* 10 (1963) 586.

[1964] On reducibility by recursive functions, *Proc. Am. Math. Soc.* 15 (1964) 889–892.

[1966a] Linear orderings under one-one reducibility, *J. Symb. Log.* 31 (1966) 70–85.

[1971] A note on 'axioms' for computational complexity and computations of finite functions, *Inf. Contr.* 19 (1971) 377–386.

[1971a] A note on dense and non dense families of complexity classes, *Math. Syst. Th.* 5 (1971) 66–70.

[1971b] Speed-up by changing the order in which sets are enumerated, *Math. Syst. Th.* 5 (1971) 148–156.

[1973] Easy constructions in complexity theory: gap and speed-up theorems, *Proc. Am. Math. Soc.* 37 (1973) 555–563.

[1977] Optimization among provably equivalent programs, *J. Ass. Comp. Mach.* 24 (1977) 693–700.

[1983] Some structural properties of polynomial reducibilities and sets in NP, *Proc. Symp. Th. Comp.* 15 (1983) 392–401.

[1990] Juris Hartmanis: fundamental contributions to isomorphism problems, in *Complexity Theory Retrospective*, Selman ed., Springer, 1990, pp. 28–58.

[1992] How reductions to sparse sets collapse the Polynomial Time Hierarchy: a primer, *S.I.G.A.C.T. News* 23 (1992) 83–93.

Zachos, S.

[1983] Collapsing probabilistic Polynomial Time Hierarchies, *Proc. Conf. Compl. Th.* (1983) 75–81.

Zachos, S., and Heller, H.

[1986] A decisive characterization of BPP, *Inf. Contr.* 69 (1986) 125–135.

Zadeh, N.

[1973] A bad network problem for the Simplex Method and other Minimum Cost Flow algorithms, *Math. Prog.* 5 (1973) 255–266.

Zak, S.

[1983] A Turing machine time hierarchy, *Theor. Comp. Sci.* 26 (1983) 327–333.

Zemke, F.

[1977] P.R.-regulated systems of notation and the subrecursive hierarchy equivalence property, *Trans. Am. Math. Soc.* 234 (1977) 89–118.

Zeugmann, T.

[1983] A-posteriori characterizations in inductive inference of recursive functions, *J. Inf. Proc. Cyb.* 19 (1983) 559–594.

[1983a] On the synthesis of fastest programs in inductive inference, *J. Inf. Proc. Cyb.* 19 (1983) 625–642.

[1991] Inductive inference of optimal programs. A survey and open problems, *Springer Lect. Not. Comp. Sci.* 543 (1991) 208–222.

Zimand, M.

[1993] If not empty, NP − P is topologically large, *Theor. Comp. Sci.* 119 (1993) 293–310.

æ

Notation Index

EX_{opt}	identification by optimal explanation 125
EX^*	identification by explanation with finitely many errors 126
EX^n	identification by explanation with at most n errors 129
BC_{cons}	behaviorally correct and consistent identification 131
BC	behaviorally correct identification 131
BC^*	behaviorally correct identification with finitely many errors 138
BC^n	behaviorally correct identification with at most n errors 140
$\leq_{EX}$, $\equiv_{EX}$	EX-reducibility 141, 142
$\boldsymbol{\mathcal{D}_{EX}}$	structure of EX-degrees 142
$\mathbf{0}_{EX}$, $\mathbf{1}_{EX}$	least and greatest EX-degrees 142
$\boldsymbol{\mathcal{R}_{EX}}$	structure of r.e. EX-degrees 142
$\boldsymbol{\mathcal{D}_{NV}}$	structure of NV-degrees 143
$\boldsymbol{\mathcal{D}_{BC}}$	structure of BC-degrees 143

Chapter VIII

REAL(F)	real time 148
(N)CONSPACE	(nondeterministic) constant space 150
(N)LOGSPACE	(nondeterministic) logarithmic space 160
$\leq_m^{\text{LOGSPACE}}$	logarithmic space m-reduction 161
$\mathcal{K}_0^{\text{NLOGSPACE}}$	complete set for NLOGSPACE 162
PF	polynomial time computable functions 164
$\sum_{y\leq\|z\|}$, $\prod_{y\leq\|z\|}$	weak bounded sum and product 166
$\forall x \leq \|y\|$, $\exists x \leq \|y\|$	weak bounded quantifiers 166
$\mu y_{\leq\|z\|}$	weak bounded μ-recursion 166
wexp_n	weak exponentials 177
d_{PF}	diagonal function for PF 177
P	polynomial time 181
P_n	time $\|x\|^n$ 181
ALOGSPACE	alternating logarithmic space 181
$\mathcal{K}_0^{\text{P}}$	complete set for P 182
POLYLOGSPACE	polylogarithmic space 185
(N)LOGSPACE^n	(nondeterministic) $\log^n$ space 185
SC, NC	Steve's and Nick's Classes 187
$\leq_m^{\text{P}}$, $\equiv_m^{\text{P}}$	polynomial time m-reducibility 188
$\boldsymbol{\mathcal{D}_m^{\text{P}}}$	structure of polynomial time m-degrees 188
$\mathbf{0}_m^{\text{P}}$	least polynomial time m-degree 188
$\leq_T^{\text{P}}$, $\equiv_T^{\text{P}}$	polynomial time T-reducibility 189
$\boldsymbol{\mathcal{D}_T^{\text{P}}}$	structure of polynomial time T-degrees 190
$\mathbf{0}_T^{\text{P}}$	least polynomial time T-degree 190
$\boldsymbol{\mathcal{D}_{hm}^{\text{P}}}$	structure of honest polynomial time m-degrees 197
NP_n	nondeterministic time $\|x\|^n$ 198
UP	polynomial time unique acceptance 203
$\mathcal{K}_0^{\text{NP}}$	complete set for NP 203
$\equiv^{\text{P}}$	polynomial time isomorphism 204
co-NP	sets with complement in NP 208
(N)P^A	(nondeterministic) polynomial time relative to the oracle A 213
$\mathcal{K}_0^{\text{NP}^A}$	complete set for NP^A 213
NP_b^A	bounded-query nondeterministic polynomial time 217
$\boldsymbol{\mathcal{R}_m^{\text{P}}}$, $\boldsymbol{\mathcal{R}_T^{\text{P}}}$	structures of polynomial time degrees of NP sets 222
$\mathcal{L}_{\text{P}}^*$, $\mathcal{A}_{\text{P}}^*$	language and structure for Bounded Arithmetic 224
$\mathcal{A}_{\text{P}}^* \models \varphi$	φ is true in $\mathcal{A}_{\text{P}}^*$ 224

Σ_n^{P}, Π_n^{P}, Δ_n^{P}	Polynomial Time Hierarchy classes 226
PH	Polynomial Time Hierarchy 227
$\mathcal{L}_{\mathrm{P}}^2$, $\mathcal{A}_{\mathrm{P}}^2$	second-order language and structure on finite domains 227
$\mathcal{A}_{\mathrm{P}}^2 \models \varphi$	φ is true in $\mathcal{A}_{\mathrm{P}}^2$ 227
$\emptyset_{\mathrm{P}}^{(n)}$	complete set for Σ_n^{P} 230
BPH	Boolean Polynomial Time Hierarchy 231
$\mathbf{L}_n^{\mathrm{P}}$, $\mathbf{H}_n^{\mathrm{P}}$, $\mathbf{I}^{\mathrm{P}}$	Polynomial Time Jump Hierarchy classes 233
PH/poly	Advice Polynomial Time Hierarchy 234
P/poly	polynomial size circuits 234
$\leq_{\Delta_n^{\mathrm{P}}}$	Δ_n^{P}-reducibility 235
$\leq^{SN}$	strong nondeterministic Turing reducibility 235
(N)LINSPACE	(nondeterministic) linear space 236
(N)PSPACE	(nondeterministic) polynomial space 237
AP	alternating polynomial space 238
(N)PSPACE$_n$	(nondeterministic) space $\lvert x\rvert^n$ 238
$\mathcal{K}_0^{\mathrm{PSPACE}}$, $\emptyset_{\mathrm{P}}^{(\omega)}$	complete sets for PSPACE 239, 243
PSPACE^A	polynomial space relative to the oracle A 245
(N)PSPACE$_b^A$	bounded-query (nondeterministic) polynomial space 246, 247
RP	random polynomial time 248
ZPP	zero-error probability polynomial time 248
BPP	bounded-error probability polynomial time 248
PP	probabilistic polynomial time 248
$\mathcal{K}_0^{\mathrm{PP}}$	complete set for PP 248
ABPP	alternating bounded-error probability polynomial time 249
APP	alternating probabilistic polynomial time 249
IP	interactive proof systems 249
(N)EXP	(nondeterministic) exponential time 250
(N)EXP$_n$	(nondeterministic) time $2^{n\cdot\lvert x\rvert}$ 250, 251
$\mathcal{K}_0^{\mathrm{(N)EXP}}$	complete sets for (N)EXP 251
POLY(N)EXP	polyexponential (nondeterministic) time 255
APSPACE	alternating polynomial space 255
POLY(N)EXP$_n$	(nondeterministic) time $2^{\lvert x\rvert^n}$ 256
MIP	multiple prover interactive proof systems 258
(N)EXPSPACE	(nondeterministic) exponential space 259
(N)EXPSPACE$_n$	(nondeterministic) space $2^{n\cdot\lvert x\rvert}$ 259
$\mathcal{K}_0^{\mathrm{EXPSPACE}}$	complete set for EXPSPACE 260
$\exp_n$	superexponential functions 261, 276
(N)EXP^n	(nondeterministic) time $\exp_n$ 261, 278
EXPSPACE^n	(nondeterministic) space $\exp_n$ 261, 278
$\mathcal{E}$	elementary functions 265
$d_{\mathcal{E}}$	diagonal function for $\mathcal{E}$ 276, 279
f_α	Slow Growing Hierarchy 280
ϵ_0	first fixed-point of the ordinal exponential function 281
$\mathcal{E}(g)$	functions elementary in g 284
$\leq_{\mathcal{E}}$, $\equiv_{\mathcal{E}}$	elementary reducibility 284
$\mathcal{P}_1$	primitive recursive functions 286
$p(a,b)$	predecessor function 289, 317
h_n	Fast Growing Hierarchy 298
$d_{\mathcal{P}_1}$	diagonal function for $\mathcal{P}_1$ 298
$\mathcal{E}_n$ $(n \geq 3)$	Grzegorczyk Hierarchy 301
$\mathcal{A}_n$ $(n \geq 2)$	Primitive Recursion Hierarchy 304

$\mathcal{E}_0$, $\mathcal{E}_1$, $\mathcal{E}_2$	low levels of the Grzegorczyk Hierarchy 306
$\mathcal{A}_0$, $\mathcal{A}_1$	low levels of the Primitive Recursion Hierarchy 306
$\mathcal{L}_n$	Loop Hierarchy 307
g_α	Moderate Growing Hierarchy 308
$\mathcal{G}_\alpha$	refined Grzegorczyk Hierarchy 310
$\phi_n(\gamma)$	Veblen Hierarchy 311
$\mathcal{P}_n$, $\mathcal{P}_\omega$	n-recursive and multiple recursive functions 315
$\prec_n$, ord_n, num_n	canonical well-ordering of ϵ_0 318
$U(\alpha)$, $N(\alpha)$	unnested and nested α-recursive functions, for $\alpha < \epsilon_0$ 319
$U(\epsilon_0)$, $N(\epsilon_0)$	unnested and nested ϵ_0-recursive functions 319
h_α	extended Fast Growing Hierarchy 326
$d_{U(\epsilon_0)}$	diagonal function for $U(\epsilon_0)$ 330
$\mathcal{E}_\alpha$	extended Grzegorczyk Hierarchy 338
$\mathcal{G}_\alpha$	extended refined Grzegorczyk Hierarchy 343
$\phi_\alpha(\gamma)$	Veblen-Bachmann Hierarchy 344

Chapter IX

$A \subseteq^* B$	A is almost contained in B 357
$A =^* B$	A and B differ finitely 357
A^*, $\mathcal{C}^*$	equivalence classes of A and $\mathcal{C}^*$ w.r.t. $=^*$ 357, 358
REC, **REC***	structures of recursive sets under inclusion 358
$\boldsymbol{\mathcal{E}}$, $\boldsymbol{\mathcal{E}^*}$	structures of r.e. sets under inclusion 368
$\mathcal{L}(A)$, $\mathcal{L}^*(A)$	principal filters generated by A 368
$\mathcal{W}_e \setminus \mathcal{W}_i$	elements generated in $\mathcal{W}_e$ before being generated in $\mathcal{W}_i$ 369
$\mathcal{W}_e \searrow \mathcal{W}_i$	elements of $\mathcal{W}_e \setminus \mathcal{W}_i$ that later turn out in $\mathcal{W}_i$ 369
$\boldsymbol{\mathcal{E}^*}(A - B)$	lattice of r.e. sets restricted to $A - B$ 379
C_B	computation function of B 381
$\boldsymbol{\mathcal{E}}^\alpha$	α-factorization of $\boldsymbol{\mathcal{E}}$ 384
$\boldsymbol{\mathcal{E}}(\overline{A})$	lattice of r.e. sets restricted to $\overline{A}$ 412
$\boldsymbol{\mathcal{E}^A}$	relativization of $\boldsymbol{\mathcal{E}}$ to A 413
H^A	Halting Problem relativized to A 426
H_A	weak jump of A 426
F^A	Finiteness Problem relativized to A 440
F_A	weak double jump of A 440
EX_s, EX_s^*	set identification by enumeration 450
BC_s, BC_s^*	behaviorally correct set identification 450

Chapter X

$\boldsymbol{\mathcal{R}_r}$, $\boldsymbol{\mathcal{R}}$	structures of r.e. degrees 456
P_e, N_e	positive and negative requirements 457
$u(e, x, s)$	use function 459
$r(e, s)$	restraint function 459–460
I_e	injury set 461
$l(e, s)$	length of agreement function 465
$m(e, s)$	modified length of agreement function 467
$\hat{u}$, $\hat{l}$, $\hat{r}$	hat versions of u, l and r 478
$\hat{m}$	hat version of m 484
$\prec$	order of strings on a priority tree 490
R_σ	requirements on a priority tree 490
$\leq_\alpha$, $\equiv_\alpha$	α-reducibility 496

Index