PERSPECTIVES IN LOGIC

Manuel Lerman

DEGREES OF UNSOLVABILITY

LOCAL AND GLOBAL THEORY

ASL

CAMBRIDGE

Degrees of Unsolvability

Since their inception, the Perspectives in Logic and Lecture Notes in Logic series have published seminal works by leading logicians. Many of the original books in the series have been unavailable for years, but they are now in print once again.

In this volume, the 11th publication in the Perspectives in Logic series, Manuel Lerman presents a systematic study of the interaction between local and global degree theory. He introduces the reader to the fascinating combinatorial methods of recursion theory while simultaneously showing how to use these methods to prove global theorems about degrees.

The intended reader will have already taken a graduate-level course in recursion theory, but this book will also be accessible to those with some background in mathematical logic and a feeling for computability. It will prove a key reference to enable readers to easily locate facts about degrees and it will direct them to further results.

Manuel Lerman works in the Department of Mathematics at the University of Connecticut.

PERSPECTIVES IN LOGIC

The *Perspectives in Logic* series publishes substantial, high-quality books whose central theme lies in any area or aspect of logic. Books that present new material not now available in book form are particularly welcome. The series ranges from introductory texts suitable for beginning graduate courses to specialized monographs at the frontiers of research. Each book offers an illuminating perspective for its intended audience.

The series has its origins in the old *Perspectives in Mathematical Logic* series edited by the Ω-Group for "Mathematische Logik" of the Heidelberger Akademie der Wissenschaften, whose beginnings date back to the 1960s. The Association for Symbolic Logic has assumed editorial responsibility for the series and changed its name to reflect its interest in books that span the full range of disciplines in which logic plays an important role.

For more information, see www.aslonline.org/books_perspectives.html

PERSPECTIVES IN LOGIC

Degrees of Unsolvability

Local and Global Theory

MANUEL LERMAN
University of Connecticut

ASSOCIATION FOR SYMBOLIC LOGIC

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

4843/24, 2nd Floor, Ansari Road, Daryaganj, Delhi – 110002, India

79 Anson Road, #06–04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107168138
10.1017/9781316717059

Association for Symbolic Logic
Richard A. Shore, Publisher
Department of Mathematics, Cornell University, Ithaca, NY 14853
http://www.aslonline.org

A catalogue record for this publication is available from the British Library.

ISBN 978-1-107-16813-8 Hardback

To Maxine,
Elliot and Sharon

Preface to the Series
Perspectives in Mathematical Logic

(Edited by the Ω-group for "Mathematische Logik" of the Heidelberger Akademie der Wissenschaften)

On Perspectives. *Mathematical logic arose from a concern with the nature and the limits of rational or mathematical thought, and from a desire to systematise the modes of its expression. The pioneering investigations were diverse and largely autonomous. As time passed, and more particularly in the last two decades, interconnections between different lines of research and links with other branches of mathematics proliferated. The subject is now both rich and varied. It is the aim of the series to provide, as it were, maps or guides to this complex terrain. We shall not aim at encyclopaedic coverage; nor do we wish to prescribe, like Euclid, a definitive version of the elements of the subject. We are not committed to any particular philosophical programme. Nevertheless we have tried by critical discussion to ensure that each book represents a coherent line of thought; and that, by developing certain* themes, *it will be of greater interest than a mere assemblage of results and techniques.*

The books in the series differ in level: some are introductory, some highly specialised. They also differ in scope: some offer a wide view of an area, others present a single line of thought. Each book is, at its own level, reasonably self-contained. Although no book depends on another as prerequisite, we have encouraged authors to fit their book in with other planned volumes, sometimes deliberately seeking coverage of the same material from different points of view. We have tried to attain a reasonable degree of uniformity of notation and arrangement. However, the books in the series are written by individual authors, not by the group. Plans for books are discussed and argued about at length. Later, encouragement is given and revisions suggested. But it is the authors who do the work; if, as we hope, the series proves of value, the credit will be theirs.

History of the Ω-Group. *During 1968 the idea of an integrated series of monographs on mathematical logic was first mooted. Various discussions led to a meeting at Oberwolfach in the spring of 1969. Here the founding members of the group (R. O. Gandy, A. Levy, G. H. Müller, G. E. Sacks, D. S. Scott) discussed the project in earnest and decided to go ahead with it. Professor F. K. Schmidt and Professor Hans Hermes gave us encouragement and support. Later Hans Hermes joined the group. To begin with all was fluid. How ambitious should we be? Should we write the books ourselves? How long would it take? Plans for authorless books were promoted, savaged and scrapped. Gradually there emerged a form and a method. At the end of an infinite discussion we found our name, and that of the series. We established our centre in Heidelberg. We agreed to meet twice a year together with authors, consultants and* *assistants, generally in Oberwolfach. We soon found the value of collaboration: on the one hand the permanence of the founding group gave coherence to the over-all plans; on the other hand the stimulus of new contributors kept the project alive and flexible. Above all, we found how intensive discussion could modify the authors' ideas and our own. Often the battle ended with a detailed plan for a better book which the author was keen to write and which would indeed contribute a perspective.*

Oberwolfach, September 1975

Acknowledgements. *In starting our enterprise we essentially were relying on the personal confidence and understanding of Professor Martin Barner of the Mathematisches Forschungsinstitut Oberwolfach, Dr. Klaus Peters of Springer-Verlag and Dipl.-Ing. Penschuck of the Stiftung Volkswagenwerk. Through the Stiftung Volkswagenwerk we received a generous grant (1970 – 1973) as an initial help which made our existence as a working group possible.*

Since 1974 the Heidelberger Akademie der Wissenschaften (Mathematisch-Naturwissenschaftliche Klasse) has incorporated our enterprise into its general scientific program. The initiative for this step was taken by the late Professor F. K. Schmidt, and the former President of the Academy, Professor W. Doerr.

Through all the years, the Academy has supported our research project, especially our meetings and the continuous work on the Logic Bibliography, in an outstandingly generous way. We could always rely on their readiness to provide help wherever it was needed.

Assistance in many various respects was provided by Drs. U. Felgner and K. Gloede (till 1975) and Drs. D. Schmidt and H. Zeitler (till 1979). Last but not least, our indefatigable secretary Elfriede Ihrig was and is essential in running our enterprise.

We thank all those concerned.

Heidelberg, September 1982

R. O. Gandy *H. Hermes*
A. Levy *G. H. Müller*
G. E. Sacks *D. S. Scott*

Author's Preface

I first seriously contemplated writing a book on degree theory in 1976 while I was visiting the University of Illinois at Chicago Circle. There was, at that time, some interest in an Ω-series book about degree theory, and through the encouragement of Bob Soare, I decided to make a proposal to write such a book. Degree theory had, at that time, matured to the point where the local structure results which had been the mainstay of the earlier papers in the area were finding a steadily increasing number of applications to global degree theory. Michael Yates was the first to realize that the time had come for a systematic study of the interaction between local and global degree theory, and his papers had a considerable influence on the content of this book.

During the time that the book was being written and rewritten, there was an explosion in the number of global theorems about the degrees which were proved as applications of local theorems. The global results, in turn, pointed the way to new local theorems which were needed in order to make further progress. I have tried to update the book continuously, in order to be able to present some of the more recent results. It is my hope to introduce the reader to some of the fascinating combinatorial methods of Recursion Theory while simultaneously showing how to use these methods to prove some beautiful global theorems about the degrees.

This book has gone through several drafts. An earlier version was used for a one semester course at the University of Connecticut during the Fall Semester of 1979, at which time a special year in Logic was taking place. Many helpful comments were received from visitors to UConn and UConn faculty at that time. Klaus Ambos, David Miller and James Schmerl are to be thanked for their helpful comments. Steven Brackin and Peter Fejer carefully read sizable portions of that version and supplied me with many corrections and helpful suggestions on presentation. Richard Shore, Stephen Simpson and Robert Soare gave helpful advice about content and presentation of material. Other people whose comments, corrections and suggestions were of great help are Richard Epstein, Harold Hodes, Carl Jockusch, Jr. Azriel Levy and George Odifreddi. I am especially grateful to David Odell who carefully read the manuscript which I expected to be the final one, and to Richard Shore who used that same manuscript for a course at Cornell University during the Fall Semester of 1981. They supplied me with many corrections and helpful suggestions on presentation of material which have been incorporated into the book and which, I hope, have greatly enhanced the readability of the book.

Also, the meetings of the Ω-group provided me with many suggestions which influenced the continuously evolving formulation of the book.

I owe a debt of gratitude to my teachers, Anil Nerode and Thomas McLaughlin, who introduced me to Recursion Theory, and to Gerald Sacks who continued my education and provided me with much needed encouragement and dubious advice. Finally, I thank my colleagues who have shown an interest in my work and have stimulated me with theirs.

Storrs, February 5, 1983 Manuel Lerman

Table of Contents

Introduction 1

Part A. The Structure of the Degrees 5

Chapter I. Recursive Functions 7

1. The Recursive and Partial Recursive Functions. 7
2. Relative Recursion. 11
3. The Enumeration and Recursion Theorems 12

Chapter II. Embeddings and Extensions of Embeddings in the Degrees . . 15

1. Uppersemilattice Structure for the Degrees 15
2. Incomparable Degrees. 17
3. Embeddings into the Degrees. 22
4. Extensions of Embeddings into the Degrees. 27

Chapter III. The Jump Operator. 34

1. The Arithmetical Hierarchy. 34
2. The Jump Operator 39
3. Embeddings and Exact Pairs Below $\mathbf{0}'$. 41
4. Jump Inversion 45
5. Maximal Antichains and Maximal Independent Sets Below $\mathbf{0}'$. . . . 52
6. Maximal Chains Below $\mathbf{0}'$. 59
7. Classes of Degrees Determined by the Jump Operation 61
8. More Exact Pairs 69

Chapter IV. High/Low Hierarchies 75

1. High/Low Hierarchies. 75
2. $\mathbf{GL}_1$ and 1-Generic Degrees. 79
3. $\mathbf{GL}_2$ and its Complement 84
4. $\mathbf{GH}_1$. 89
5. Automorphism Bases 94

Part B. Countable Ideals of Degrees 99

Chapter V. Minimal Degrees. 101

1. Binary Trees 101
2. Minimal Degrees. 104
3. Double Jumps of Minimal Degrees 108
4. Minimal Covers and Minimal Upper Bounds. 111
5. Cones of Minimal Covers. 113

Chapter VI. Finite Distributive Lattices. 118

1. Usl Representations 118
2. Uniform Trees 124
3. Splitting Trees 129
4. Initial Segments of $\mathscr{D}$ 135
5. An Automorphism Base for $\mathscr{D}$ 139

Chapter VII. Finite Lattices 148

1. Weakly Homogeneous Sequential Lattice Tables 148
2. Uniform Trees 150
3. Splitting Trees 151
4. Finite Ideals of $\mathscr{D}$ 156

Chapter VIII. Countable Usls 161

1. Countable Ideals of $\mathscr{D}$. 161
2. Jump Preserving Isomorphisms. 165
3. The Degree of Th($\mathscr{D}$) 168
4. Elementary Equivalence over $\mathscr{D}'$ 174
5. Isomorphisms Between Cones of Degrees 175

Part C. Initial Segments of $\mathscr{D}$ and the Jump Operator 181

Chapter IX. Minimal Degrees and High/Low Hierarchies. 183

1. Partial Recursive Trees 183
2. Minimal Degrees Below $\mathbf{0}'$ 185
3. Minimal Degrees Below Degrees in $\mathbf{GH}_1$ 192

Chapter X. Jumps of Minimal Degrees 198

1. Targets 198
2. Jumps of Minimal Degrees 206

Chapter XI. Bounding Minimal Degrees with Recursively Enumerable Degrees 212

1. Trees Permitted by Recursively Enumerable Sets 212
2. Minimal Degrees and Recursively Enumerable Permitting 215

Chapter XII. Initial Segments of $\mathscr{D}[\mathbf{0}, \mathbf{0}']$ 221

1. Weakly Uniform Trees 221
2. Subtree Constructions 226
3. Splitting Trees 234
4. The Construction 243
5. Generalizations and Applications 256

Appendix A. Coding into Structures and Theories 262

1. Degrees of Presentations of Lattices 262
2. Interpreting Theories within Other Theories. 264
3. Second Order Arithmetic 271

Appendix B. Lattice Tables and Representation Theorems 273

1. Finite Distributive Lattices 273
2. Finite Lattices 277
3. Countable Uppersemilattices 281

References 293

Notation Index 298

Subject Index 302

Introduction

Degree theory, as it is studied today, traces its development back to the fundamental papers of Post [1944] and Kleene and Post [1954]. These papers introduced algebraic structures which arise naturally from the classification of sets of natural numbers in terms of the amount of additional oracular information needed to compute these sets. Thus we say that *A is computable from B* if there is a computer program which identifies the elements of *A*, using a computer which has access to an oracle containing complete information about the elements of *B*.

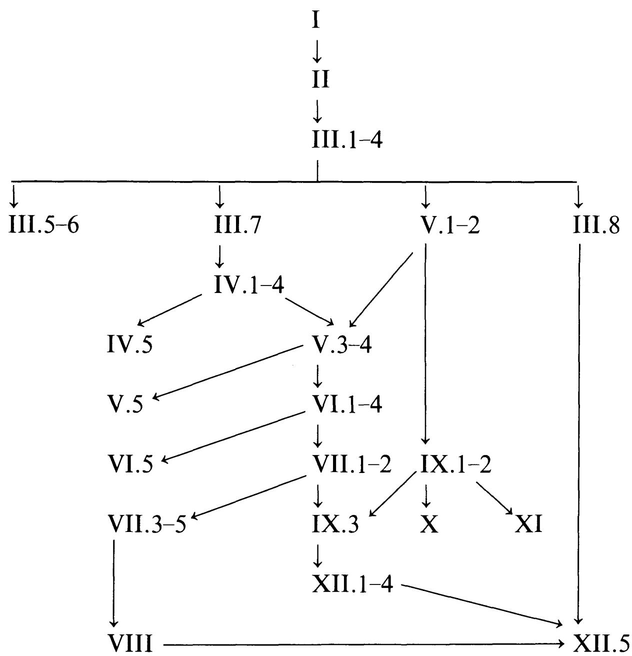
The idea of comparing sets in terms of the amount of information needed to compute them has been extended to notions of computability or constructibility which are relevant to other areas of Mathematical Logic such as Set Theory, Descriptive Set Theory, and Computational Complexity as well as Recursion Theory. However, the most widely studied notion of degree is still that of *degree of unsolvability* or *Turing degree*. The interest in this area lies as much in the fascinating combinatorial proofs which seem to be needed to obtain the results as in the attempt to unravel the mysteries of the structure. An attempt is made, in this book, to present a study of the degrees which emphasizes the methods of proof as well as the results. We also try to give the reader a feeling for the usefulness of local structure theory in determining global properties of the degrees, properties which deal with questions about homogeneity, automorphisms, decidability and definability.

This book has been designed for use by two groups of people. The main intended audience is the student who has already taken a graduate level course in Recursion Theory. An attempt has been made, however, to make the book accessible to the reader with some background in Mathematical Logic and a good feeling for computability. Chapter 1 has been devoted to a summary of basic facts about computability which are used in the book. The reader who is intuitively comfortable with these results should be able to master the book. The second intended use for the book is as a reference to enable the reader to easily locate facts about the degrees. Thus the reader is directed to further results which are related to those in a given section whenever the treatment of a topic within a section and its exercises is not complete.

The material which this book covers deals only with part of Classical Recursion Theory. A major omission is the study of the lattice of recursively enumerable sets, and the study of the recursively enumerable degrees is only cursory. These areas are normally covered in a first course in Recursion Theory, and the books of Soare [1984], Shoenfield [1971] and Rogers [1967] are recommended as sources for this material.

The book contains more material than can be covered in a one semester course. If time is short, it is advisable to sample material in some of the sections rather than cover whole sections. Sample courses for one semester would contain a core consisting of Chaps. I–V and Chap. IX, with the remaining time spent either on Chaps. VI–VIII (perhaps skipping some of the structure results, and either assuming them for the purposes of the applications of Chap. VIII, or using the exercises at the end of Chap. VI to replace the structure results of Chap. VIII in those applications), or on Chaps. X and XI. Chapter XII is best left to the reader to puzzle through on his own. The material in the appendices may be covered immediately before the section where it is used, but it is recommended that this material be left to the reader.

The following chart describes the major dependencies of one section on another within the book.



Some proofs are left unfinished, to be worked out by the reader. This is done either to avoid repeating a proof which is similar to one already presented, or when straightforward details remain to be worked out. Hints are provided for the more difficult exercises, along with references to the original papers where these results appeared. Exercises which are used later in the text have been starred.

Although an attempt has been made to be accurate in the attribution of results, it is inevitable that some omissions and perhaps errors occur. We apologize in advance for those unintentional errors.

Theorems, definitions, etc. are numbered and later referred to by chapter, section, and number within the section. Thus VI.1.2 is the numbered paragraph in

Sect. 1 of Chap. VI with number 1.2. If the reference to this paragraph is within Chap. VI, we refer to the paragraph as 1.2, dropping the VI. There are two appendices, A and B, and a reference to A.1.2 is a reference to paragraph 1.2 of Appendix A.

Definitions and Notation. The following definitions and notation will be used without further comment within the book.

Sets will be determined by listing their elements as $\{a_0, a_1, \ldots\}$ or by specification as the set of all x satisfying property P, denoted by $\{x : P(x)\}$. If A and B are sets, then we write $x \in A$ for *x is an element of A* and $A \subseteq B$ for *A is a subset of B*. We use $A \subset B$ to denote $A \subseteq B$ but $A \neq B$ (placing / through a relation symbol denotes that the relation fails to hold for the specified elements). $A \cup B$ is the *union* of A and B, i.e., the set of all elements which appear either in A or in B, and $A \cap B$ denotes the *intersection* of A and B, i.e., the set of all elements which appear in both A and B. The *difference* of A and B is denoted by $A - B$ and consists of those elements which lie in A but not in B. The *symmetric difference* of A and B is denoted by $A \triangle B = (A - B) \cup (B - A)$. We will denote the *maximum* or greatest element of the partially ordered set $\langle A, \leqslant \rangle$ by $\max(A)$, and the *minimum* or least element of this set by $\min(A)$ if such maximum and/or minimum elements exist.

Let A, B and C be sets. The *cartesian product* of A and B, $A \times B$, is the set of all ordered pairs $\langle x, y \rangle$ such that $x \in A$ and $y \in B$. The cartesian product operation can be iterated, so that $A \times B \times C$ is used to denote $(A \times B) \times C$. We use A^k to denote the cartesian product of k copies of A (which is the same as the set of all k-tuples of elements of A) and $A^{<\omega}$ to denote the set of all finite sequences of elements of A. If $\bar{x} = \langle x_1, \ldots, x_k \rangle$ is a k-tuple, we use $\bar{x}^{[k]}$ to denote x_k, the kth coordinate of $\bar{x}$. Given $S \subseteq A \times B$ and $i \in A$, we use $S^{[i]}$ to denote $\{x \in B : \langle i, x \rangle \in S\}$.

We use $\emptyset$ to denote the empty set, and N to denote the set of natural numbers $\{0, 1, \ldots\}$. Given $A, B \in N$, we denote the *direct sum* of A and B by $A \oplus B = \{2x : x \in A\} \cup \{2x + 1 : x \in B\}$. For any set A, $|A|$ will denote the cardinality of A. The infinite cardinal numbers are $\aleph_0, \aleph_1, \ldots$ in order, and $2^{\aleph_0}$ is the cardinality of the continuum.

A *partial function* φ from A to B (written $\varphi : A \to B$) is a subset of the set of ordered pairs $\{\langle x, y \rangle : x \in A \,\&\, y \in B\}$ such that for each $x \in A$ there is at most one $y \in B$ such that $\langle x, y \rangle \in \varphi$. We write $\varphi(x)\downarrow$ ($\varphi(x)$ *converges*) for $\langle x, y \rangle \in \varphi$, and $\varphi(x)\uparrow$ ($\varphi(x)$ *diverges*) if $x \in A$ and for all $y \in B$ $\langle x, y \rangle \notin \varphi$. We will sometimes denote the function φ with the notation $x \mapsto \varphi(x)$. The *domain* of φ is denoted by $\mathrm{dom}(\varphi) = \{x \in A : \varphi(x)\downarrow\}$ and B is called the range of φ, denoted by $\mathrm{rng}(\varphi)$. If $\mathrm{dom}(\varphi) = A$, we call φ a *total function*. The word total, however, will frequently be dropped. Thus unless otherwise specified, a function will always be total. In general, we use the lower case Roman letters $f, g, h, \ldots$ to denote functions with domain N and lower case Greek letters φ, ψ, $\theta, \ldots$ to denote partial functions with domain $\subseteq N$. The corresponding upper case letters are reserved for *functionals*, i.e., maps taking functions into functions. A set S is identified with its *characteristic function* χ_S where $\chi_S(x) = 1$ if $x \in A$ and $\chi_S(x) = 0$ otherwise. If φ is a partial function and $B \subseteq \mathrm{dom}(\varphi)$, then $\varphi \restriction B$ is the *restriction* of φ to B, i.e., the function with domain B which agrees with φ on B. By the previous definition, the restriction notation applies to sets as well as to functions.

Given $f: N \to N$, we write $\lim_s f(s) = y$ if $\{s: f(s) \neq y\}$ is finite, and $\lim_s f(s) = \infty$ if, for every $y \in N$, $\{s: f(s) = y\}$ is finite. We write $\limsup_s f(s) = y$ if $\{s: f(s) = y\}$ is infinite and $\{s: f(s) > y\}$ is finite; and $\limsup_s f(s) = \infty$ if, for every $y \in N$, $\{s: f(s) \geqslant y\}$ is infinite. We write $\liminf_s f(s) = y$ if $\{s: f(s) = y\}$ is infinite and $\{s: f(s) < y\}$ is finite; and $\liminf_s f(s) = \infty$ if, for every $y \in N$, $\{s: f(s) \leqslant y\}$ is finite. If $\{\alpha_s : s \in N\}$ is a sequence of finite sequences of integers, then we write $\lim_s \alpha_s$ for the partial function θ such that for all $x \in N$, $\theta(x)\downarrow$ if and only if $\lim_s \alpha_s(x)\downarrow$, in which case $\theta(x) = \lim_s \alpha_s(x)$. Given two sequences of integers α and β, we say that α *lexicographically precedes* β if either $\alpha \subset \beta$ or $\alpha(x) < \beta(x)$ for the least x such that $\alpha(x) \neq \beta(x)$. We write $\limsup_s \alpha_s = \theta$ if θ is a sequence of integers and for all $x \in N$, $\{s: \alpha_s \restriction x = \theta \restriction x\}$ is infinite and $\{s: \theta$ lexicographically precedes $\alpha_s\}$ is finite. We write $\liminf_s \alpha_s = \theta$ for θ as in the preceding sentence if $\{s: \alpha_s \restriction x = \theta \restriction x\}$ is infinite for each $x \in N$, and $\{s: \alpha_s \not\subseteq \theta$ and α_s lexicographically precedes $\theta\}$ is finite.

We use Church's lambda notation to define new functions from old ones. If $f(x_1, \ldots, x_n, y_1, \ldots, y_k)$ is a function of $n+k$ variables, then $\lambda x_1 \cdots x_n f(x_1, \ldots, x_n, y_1, \ldots, y_k)$ denotes the function g of n variables defined by $g(x_1, \ldots, x_n) = f(x_1, \ldots, x_n, y_1, \ldots, y_k)$.

If φ and ψ are partial functions, then we write $\varphi \subseteq \psi$ (ψ *extends* φ) if $\mathrm{dom}(\varphi) \subseteq \mathrm{dom}(\psi)$ and for all $x \in \mathrm{dom}(\varphi)$, $\psi(x)\downarrow = \varphi(x)$. We say that φ and ψ are incomparable and write $\varphi \mid \psi$ if neither $\varphi \subseteq \psi$ or $\psi \subseteq \varphi$.

We write A^B for the set of all functions from B into A. Since sets are identified with their characteristic functions, 2^S then denotes the *power set* of S, i.e., the set of all subsets of S.

Standard interval notation will be used for sets A partially ordered by $\leqslant$. Thus $[a, b]$ will denote $\{x \in A : a \leqslant x \leqslant b\}$, (a, b) will denote $\{x : a < x < b\}$, (a, ∞) will denote $\{x : x \geqslant a\}$, $(-\infty, b]$ will denote $\{x : x \leqslant b\}$, etc. Structures will be denoted by $\mathscr{A} = \langle A, R_0, \ldots, R_n, f_0, \ldots, f_k, c_0, \ldots, c_m \rangle$ where A is the *universe* of $\mathscr{A}$, $R_0, \ldots, R_n$ are relations on cartesian products of A, $f_0, \ldots, f_k$ are functions from cartesian products of A into A, and $c_0, \ldots, c_m$ are designated elements of A. The partially ordered set above is thus denoted by $\mathscr{A} = \langle A, \leqslant \rangle$.

We will use the logical symbols & to denote *and*, $\vee$ to denote *or*, $\neg$ to denote *not*, $\to$ to denote *implies*, and $\leftrightarrow$ to denote *if and only if*. $\exists$ will denote the *existential quantifier* and $\forall$ will denote the *universal quantifier*. We will write $\bigwedge_{i=0}^{n} \sigma_i$ for $\sigma_0 \,\&\, \sigma_1 \,\&\, \cdots \,\&\, \sigma_n$ and $\bigvee_{i=0}^{n} \sigma_i$ to denote $\sigma_0 \vee \sigma_1 \vee \cdots \vee \sigma_n$. When we are not using a formal language, we will use $\Rightarrow$ and $\Leftrightarrow$ in place of $\to$ and $\leftrightarrow$ respectively.

$\blacksquare$ will denote the end of a proof.

Part A

The Structure of the Degrees

Chapter I

Recursive Functions

This chapter is introductory in nature. We summarize material which is normally covered in a first course in Recursion Theory and which will be assumed within this book. Recursive and partial recursive functions are introduced and Church's Thesis is discussed. Relative recursion is then defined, and the Enumeration and Recursion Theorems are stated without proof. The reader familiar with this material should quickly skim through the chapter in order to become familiar with our notation. We refer the reader to the first five chapters of Cutland [1980] for a careful rigorous treatment of the material introduced in this chapter.

1. The Recursive and Partial Recursive Functions

The search for algorithms has pervaded Mathematics throughout its history. It was not until this century, however, that rigorous mathematical definitions of *algorithm* were discovered, giving rise to the class of partial recursive functions.

This book deals with a classification of total functions of the form $f: N \to N$ in terms of the information required to compute such a function. The rules for carrying out such computations are *algorithms* (partial functions $\varphi: N^k \to N$ for some $k \in N$) with access to information possessed by *oracles*. The easiest functions to compute are those for which no oracular information is required, the recursive functions. Thus we begin by defining the (total) recursive functions, and then indicate how to modify this definition to obtain the class of partial recursive functions. The section concludes with discussions of Church's Thesis and of general *spaces* on which recursive functions can be defined.

1.1 Definition. Let $R \subseteq N^{k+1}$. $\mu y[(x_1, \ldots, x_k, y) \in R]$ is the least y such that $(x_1, \ldots, x_k, y) \in R$ if such a y exists, and is undefined otherwise. Henceforth, we will refer to μ as the *least number operator*.

1.2 Definition. The class $\mathscr{R}$ of *recursive functions* is the smallest class of functions with domain N^k for some $k \in N$ and range N which contains:

(i) The *zero function*: $Z(x) = 0$ for all $x \in N$;
(ii) The *successor function*: $S(x) = x + 1$ for all $x \in N$;

(iii) The *projection functions*: $P_j^n(x_0, \ldots, x_n) = x_j$ for all $n, x_0, \ldots, x_n \in N$ and $j \leqslant n$;

and is closed under:

(iv) *Substitution*: For all m, $k \in N$, if all of $g(x_0, \ldots, x_m)$, $h_0(y_0, \ldots, y_k), \ldots,$ $h_m(y_0, \ldots, y_k)$ are elements of $\mathscr{R}$, then

$$f(y_0, \ldots, y_k) = g(h_0(y_0, \ldots, y_k), \ldots, h_m(y_0, \ldots, y_k))$$

is an element of $\mathscr{R}$;

(v) *Recursion*: For all $n \in N$, if $g(x_0, \ldots, x_n)$ and $h(x_0, \ldots, x_{n+2})$ are elements of $\mathscr{R}$, then $f(x_0, \ldots, x_{n+1})$ is an element of $\mathscr{R}$, where

$$f(x_0, \ldots, x_n, 0) = g(x_0, \ldots, x_n)$$

and

$$f(x_0, \ldots, x_n, y + 1) = h(x_0, \ldots, x_n, y, f(x_0, \ldots, x_n, y));$$

(vi) The *least number operator*: For all $n \in N$, if $g(x_0, \ldots, x_n, y)$ is an element of $\mathscr{R}$ and $\forall x_0, \ldots, x_n \exists y[g(x_0, \ldots, x_n, y) = 1]$ then

$$f(x_0, \ldots, x_n) = \mu y[g(x_0, \ldots, x_n, y) = 1]$$

is an element of $\mathscr{R}$.

An element of $\mathscr{R}$ is called a *recursive function.*

1.3 Definition. Fix $n \in N$ and let $\mathscr{C}$ be a countable class of partial functions of n natural number variables. An *enumeration* of $\mathscr{C}$ is a partial function $\varphi: N^{n+1} \to N$ which lists the elements of $\mathscr{C}$, i.e.,

(i) $\qquad \forall \psi \in \mathscr{C} \, \exists k \in N(\lambda x_1, \ldots, x_n \varphi(k, x_1, \ldots, x_n) = \psi(x_0, \ldots, x_n))$

and

(ii) $\qquad \forall k \in N(\lambda x_1, \ldots, x_n \varphi(k, x_1, \ldots, x_n) \in \mathscr{C}).$

1.4 Example. Let $\mathscr{C} = \{f_i : i \in N\}$ where $f_i(x) = i$ for all $x \in N$. Then $g: N^2 \to N$ defined by $g(n, x) = n$ is an enumeration of $\mathscr{C}$.

The Enumeration Theorem for partial recursive functions of one variable is an important tool used in almost every proof in this book. What we would like to have is a recursive enumeration of the class of recursive functions of one variable. Unfortunately, such an enumeration does not exist (see Exercise 1.10). All that is needed, however, is a *partial recursive* enumeration of the class of *partial recursive* functions. With this in mind, we now introduce the class of partial recursive functions.

1.5 Remark. The obstacle to obtaining a recursive enumeration of the class of recursive functions of one variable lies in 1.2(vi), the application of the least number operator to obtain new recursive functions. There is no algorithm which will identify whether or not $\forall x_0, \ldots, x_n \exists y[g(x_0, \ldots, x_n, y) = 1]$. This difficulty can be circumvented by producing an algorithm which assigns natural numbers (called *Gödel numbers*) to computations carried out in 1.2(i)–(vi). One then searches for the least numbered computation which yields $g(x_0, \ldots, x_n, y) = 1$ for some y, say $y = y_0$, and defines

$$f(x_0, \ldots, x_n) = \begin{cases} y_0 & \text{if } y_0 \text{ is ever found,} \\ \uparrow & \text{otherwise.} \end{cases}$$

Such a procedure was carried out by Kleene, giving rise to the class of *partial recursive functions*, $\mathscr{P}$. This class contains all the recursive functions, together with some additional functions, none of which are total.

During the 1930's and 1940's, several attempts were made to give a rigorous mathematical definition of *algorithm*. One of these definitions was the class of partial recursive functions described in Remark 1.5. All of the definitions were eventually shown to be equivalent, and the equivalence of some of the early definitions prompted Church to propose his thesis, which asserts:

1.6 Church's Thesis. A function is partial recursive if and only if there is an algorithm which computes the function on its domain, and diverges outside the domain of the function.

Church's Thesis asserts that the intuitive notion of *algorithm* is equivalent to the mathematically precise notion of *partial recursive function*. The thesis is almost universally accepted, and its use has become general mathematical practice. We will be using Church's Thesis freely and without any explicit warning throughout this book, by describing the computation of a function and automatically assuming that the resulting function is partial recursive. A rigorous proof could be given in every case, but would be very tedious.

In this age of digital computers, the reader might feel most comfortable with the following description of partial recursive functions. A function f is partial recursive if there is a program for a digital computer (no restrictions on memory size are placed on such a computer, so that we assume that the computer has available to it an infinite supply of memory space, only finitely much of which is used at a given time) such that whenever x is fed as input to the computer, the computer will spew out $f(x)$ after spending a finite amount of time performing computations as directed by the program (no restrictions, however, are placed on the amount of time available) if $x \in \operatorname{dom}(f)$, and the computer will give no answer (perhaps computing forever) if $x \notin \operatorname{dom}(f)$.

To this point, we have only considered functions from N^k into N for some $k > 0$. Shoenfield [1971] has noted that N^k and N can be replaced by any *spaces*, i.e., domains which can *effectively* be placed in one-one correspondence with N. Henceforth, any space will be acceptable as either the domain or range of a recursive function. Typical spaces which we will be using later are mentioned in the next example.

1.7 Example. The following are spaces:

(i) N^k, the set of all k-tuples of natural numbers, for all $k \geqslant 1$.

(ii) $N^{<\omega}$, the set of all finite sequences of natural numbers. Henceforth we will denote $N^{<\omega}$ by $\mathcal{S}$, and call an element of $\mathcal{S}$ a *string*.

(iii) $\mathcal{S}_f = \{\sigma \in \mathcal{S} : \sigma(n) < f(n)$ for all $n \in N$ such that $\sigma(n)\downarrow\}$, where $f: N \to N$ is a recursive function and $f(x) \neq 0$ for all $x \in N$. $\mathcal{S}_f$ is called the space of *f-valued strings*. If f is the constant function $f(x) = c$ for all x, then $\mathcal{S}_c$ will be used in place of $\mathcal{S}_f$. Thus $\mathcal{S}_2$ is the space of all finite sequences of 0s and 1s.

The following notation will be used:

1.8 Definition. Let σ, $\tau \in \mathcal{S}$ be given. We say that $\sigma \subseteq \tau$ if for all $i \in N$, if $\sigma(i)\downarrow$ then $\tau(i)\downarrow$ and $\sigma(i) = \tau(i)$. Given $f: N \to N$, we say that $\sigma \subseteq f$ if for all $i \in N$, if $\sigma(i)\downarrow$ then $\sigma(i) = f(i)$. $\subset$ will denote $\subseteq$ and $\neq$. $\mathrm{lh}(\sigma) = |\{i : \sigma(i)\downarrow\}|$ is the *length* of σ. $\sigma * \tau$ is the string of length $\mathrm{lh}(\sigma) + \mathrm{lh}(\tau)$ defined by

$$\sigma * \tau(x) = \begin{cases} \sigma(x) & \text{if} \quad x < \mathrm{lh}(\sigma), \\ \tau(x - \mathrm{lh}(\sigma)) & \text{if} \quad \mathrm{lh}(\sigma) \leqslant x < \mathrm{lh}(\sigma) + \mathrm{lh}(\tau). \end{cases}$$

In the next section, we will discuss relative recursiveness. We will then be able to classify arbitrary functions $f: N \to N$ on the basis of how much additional information is required (from an oracle) in order to compute f.

1.9–1.13 Exercises

1.9 The class of *primitive recursive functions* is the smallest class of functions containing the functions mentioned in 1.2(i)–(iii) and closed under the operations of 1.2(iv)–(v).

(i) Show that there is a recursive enumeration of the class of primitive recursive functions of one variable. (*Hint*: Recursively assign Gödel numbers to computations, and define the enumeration $F(e, x)$, where $\lambda x F(e, x)$ is the function with Gödel number e.)

(ii) Show that there is a recursive function which is not primitive recursive. (*Hint*: Diagonalize against an enumeration of the primitive recursive functions.)

1.10 Show that there is no recursive enumeration of the class of recursive functions. (*Hint*: If there were such an enumeration, a diagonalization as in 1.9(ii) would produce a contradiction.)

1.11 Explain why your proof of 1.10 will not generalize to show that there is no partial recursive enumeration of the class of partial recursive functions.

1.12 Let S be a space.

(i) Show that S^n is a space.

(ii) Show that $S^{<\omega}$ is a space.

(iii) Show that there is a $T \subseteq S$ such that T is not a space.

(*Hint*: Use a cardinality argument after showing that there are only countably many algorithms.)

1.13 Let S and T be spaces. Show that:

(i) $S \times T$ is a space.

(ii) For all $n \in N$, $[0, n] \times S$ and $S \times [0, n]$ are spaces.

2. Relative Recursion

Recursion Theory classifies total functions on the basis of how much additional information must be provided by an oracle to compute the given function. This classification relies on the notion of relative recursion.

Relative recursion is defined by expanding the class of initial functions in Definition 1.1. We will use this notion in Chap. 2 to form an algebraic structure from $\{f: N \to N\}$.

2.1 Definition. Let $f: N^m \to N$ be given. The class $\mathscr{R}_f$ of *functions recursive in* f is the smallest class of functions containing f and the functions mentioned in 1.2(i)–(iii) and closed under the operations of 1.2(iv)–(vi). An element g of $\mathscr{R}_f$ is said to be *recursive in* f, written $g \leqslant_T f$.

Recursiveness in f, or relative recursion, was introduced by Turing [1939] whose name gave rise to the T in $\leqslant_T$. $\leqslant_T$ is frequently referred to as *Turing reducibility*.

The partial recursive functions, as we indicated earlier, are those partial functions which an idealized digital computer can compute. The domain of such a function is the set of numbers which, when fed as input to the computer, will eventually cause the computer to output a number.

Given $f: N^k \to N$, the *partial functions computable from* f can similarly be described through the use of a digital computer with access to the *oracle* f. The notion of computer program is generalized to allow instructions of the form *if* $f(x_1, \ldots, x_k) = y$ *proceed to a certain instruction*; *otherwise, proceed to another specified instruction*. As we proceed through the computation of a partial function g computable from f to which z has been fed as input, whenever we reach a step in the program of the type just described above, the computer does something non-constructive; it asks the f oracle whether $f(x_1, \ldots, x_k) = y$, and once the answer is received from the f oracle, continues the computation utilizing the information provided by the oracle. Complete knowledge of f allows us to compute $g(z)$ whenever $g(z)\downarrow$, so g is computable from f. Futhermore, if $g(z)\downarrow$, then since any computation is completed in finitely many steps, only a finite amount of information about f is used.

2.2 Remark. There is also a version of Church's Thesis for relative recursion which will be used freely throughout this book. It asserts that the partial functions computable from f are exactly those for which there exists an algorithm using an f oracle as above to perform the computation.

It will frequently be more convenient to use sets than functions when discussing relative recursiveness. This is easily accomplished by identifying a set with its characteristic function, which we define as follows:

2.3 Definition. Let $A \subseteq N$. The *characteristic function* of A, χ_A, is defined by

$$\chi_A(n) = \begin{cases} 0 & \text{if } x \notin A \\ 1 & \text{if } x \in A. \end{cases}$$

More generally, a relation $R \subseteq N^k$ will be identified with its *characteristic function*

χ_R, defined by

$$\chi_R(x_1, \ldots, x_k) = \begin{cases} 0 & \text{if} \quad R(x_1, \ldots, x_k) \text{ is false,} \\ 1 & \text{if} \quad R(x_1, \ldots, x_k) \text{ is true.} \end{cases}$$

2.4 Definition. A relation R is said to be *recursive in* the function f if χ_R is recursive in f.

2.5 Remarks. A summary of the history of relative recursion appears in Kleene and Post [1954]. Three versions appear in the literature, all of which were later proved to be equivalent. They were formulated by Turing [1939] generalizing the machines introduced by Turing [1937]; Kleene [1943] extending the definition of recursive functions; and Post [1948] extending the concept of canonical sets which was introduced in Post [1943].

2.6 Exercise. Let $f, g: N \to N$ be given such that $\{i : f(i) \neq g(i)\}$ is finite. Show that $f \leqslant_T g$.

3. The Enumeration and Recursion Theorems

Two basic theorems of Recursion Theory are stated in this section. The first of these, the Enumeration Theorem, will be used in virtually every proof in this book. The Recursion Theorem will be used to enable us to simplify proofs of certain theorems. Complete proofs or detailed sketches of the proofs of these theorems can be found in Soare [1984], Rogers [1967], Cutland [1980] and Kleene [1952]. A nice proof of the Recursion Theorem can also be found in Owings [1973].

The Enumeration Theorem asserts the existence of a function $\varphi(\sigma, e, x, s)$ which uniformly induces a whole class of enumerations. The definition of $\varphi(\sigma, e, x, s)$ reflects the following intuition. A computer is programmed, with e coding the program. The input x is then fed to the computer, and the computer then performs s steps as directed by the program. During these s steps, the computer may come across program instructions of the form *if* $f(x) = y$ *proceed to a certain instruction, and if* $f(x) \neq y$ *proceed to another specified, but different instruction.* When faced with such a choice, the computer asks "is $\sigma(x) = y$?". If $x \geqslant \mathrm{lh}(\sigma)$, then there will be no output. If $x < \mathrm{lh}(\sigma)$, then σ answers the question for f and the computation continues. If there is no output from the computer after s steps, the computation ceases and $\varphi(\sigma, e, x, s)\uparrow$. If a number is outputted by the end of the sth step, $\varphi(\sigma, e, x, s)\downarrow$ and is set equal to this output.

3.1 Enumeration Theorem. *There is a partial recursive function* $\varphi : \mathscr{S} \times N^3 \to N$ *with the following properties*:

(i) (*Use property*)

$$\forall \sigma, \tau \in \mathscr{S}\, \forall e, x, s, y \in N(\sigma \subseteq \tau \;\&\; \varphi(\sigma, e, x, s)\downarrow = y \to \varphi(\tau, e, x, s)\downarrow = y),$$

i.e., if a number is given as output, then oracle information extending the original information will not alter the output.

(ii) (*Permanence property*)

$$\forall \sigma \in \mathcal{S}\, \forall e, x, s, t, y \in N(s \leqslant t \,\&\, \varphi(\sigma, e, x, s)\!\downarrow\; = y \rightarrow \varphi(\sigma, e, x, t)\!\downarrow\; = y),$$

i.e., once a number is given as output, additional steps do not change this output.

(iii) (*Uniform enumeration property*) *Given* $f: N \rightarrow N$ and $\theta: N \rightarrow N$ *computable from* f, *then there is an* $e \in N$ *such that for all* x, $y \in N$

$$\theta(x) = y \leftrightarrow \exists \sigma\, \exists s(\sigma \subseteq f \,\&\, \varphi(\sigma, e, x, s)\!\downarrow\; = y),$$

i.e., for every partial function θ *computable from* f, *there is a program coded into* φ *which computes* θ. (*Note that if* θ *is not given but rather defined by the above formula, then* θ *is computable from* f.)

(iv) (*Recursiveness property*) *The domain of* φ *is a recursive subset of* $\mathcal{S} \times N^3$ (*since* s *bounds the length of a permissible computation*).

(v) (*Uniform coding property*) *Given a sequence* $\{\theta_i : i \in N\}$ *of functions computable from* $f: N \rightarrow N$ *such that the definition of* θ_i *is given by a finite set of instructions using parameter* i, *then there is a recursive function* g *such that for all* $x, y \in N$

$$\theta_i(x) = y \leftrightarrow \exists \sigma\, \exists s(\varphi(\sigma, g(i), x, s)\!\downarrow\; = y \,\&\, \sigma \subseteq f),$$

i.e., if the definitions of a class of functions are given uniformly recursively, then there is a recursive function which gives codes for programs computing each of these functions.

3.2 Remark. Let φ be the function given by the Enumeration Theorem. For all $\theta: N \rightarrow N$, define the functional Φ^θ by

(i) $$\Phi^\theta(e, x) = y \Leftrightarrow \exists \sigma \in \mathcal{S}(\sigma \subseteq \theta \,\&\, \varphi(\sigma, e, x, \mathrm{lh}(\sigma))\!\downarrow\; = y).$$

Then Φ^θ is computable from θ, and Φ^f provides an enumeration of $\mathcal{R}_f$, uniformly in f. In particular, if f is any recursive function, then Φ^f is a partial recursive enumeration of the class of partial recursive functions. If $f = \emptyset$, then we write Φ in place of Φ^f.

For the remainder of this book, φ will denote the function given by the enumeration theorem, and Φ^θ will be defined as in 3.2(i). For each $e \in N$, we will also fix the function $\Phi_e^\theta = \lambda x \Phi^\theta(e, x)$.

Property 3.1(v) is known as the *s-m-n* Theorem. Another way of stating this theorem is as follows: Let $h(x, y)$ be a function computable from f. Then there is a function g computable from f such that for all $x \in N$, $h(x, y) = \Phi_{g(x)}^f$. Thus, for example, if $h(x, y) = x$, then there is a recursive function g such that for each $x \in N$, $g(x)$ is an index for the constant function x as a recursive function.

The Recursion Theorem is a basic theorem about the enumerations mentioned in Remark 3.2. It is frequently referred to as the Fixed Point Theorem.

3.3 Recursion Theorem. *Let the recursive function* $h: N \rightarrow N$ *be given. Then there is an* $e \in N$ *such that* $\Phi_e^f = \Phi_{h(e)}^f$ *for all* $f: N \rightarrow N$.

3.4 Definition. If $\Phi_e^f = \Phi_{h(e)}^f$ then Φ_e^f is called a *fixed point* (of the enumeration Φ^f) for h.

We will be using the Recursion Theorem in the following way. Given a function $f: N \to N$ and $e \in N$, we will start with the partial function Φ_e^f and, uniformly in e, we will recursively construct a partial function $\Phi_{h(e)}^f$. An application of the Recursion Theorem will allow us to choose an e such that our starting function Φ_e^f and our constructed function $\theta_{h(e)}^f$ are identical. Thus in certain situations, the Recursion Theorem allows us to construct a function while simultaneously using information about the function in its construction. By the uses of the Recursion Theorem, the information used about the function will have to be specified at an earlier stage, although this fact is hidden in the actual applications.

3.5 Remark. The Enumeration and Recursion Theorems were discovered by Kleene (see Kleene [1952]).

3.6–3.8 Exercises. The definitions in 3.6 and 3.7 describe recursive procedures which define one partial recursive function in terms of another. For each definition, apply the Recursion Theorem to obtain a fixed point. Is this fixed point a total function? What is the fixed point?

3.6
$$\Phi_{h(e)}(n) = \begin{cases} 0 & \text{if } \ n = 0, \\ \Phi_e(n-1) & \text{if } \ n > 0 \,\&\, \Phi_e(n-1)\downarrow, \\ \uparrow & \text{otherwise.} \end{cases}$$

3.7
$$\Phi_{h(e)}(n) = \begin{cases} n & \text{if } \ \Phi_e(n)\downarrow \neq n, \\ \uparrow & \text{otherwise.} \end{cases}$$

3.8 Prove the Recursion Theorem. (*Hint*: Given $m \in N$, define

$$\psi(m, x) = \begin{cases} \Phi_{\Phi_m(m)}(x) & \text{if } \ \Phi_m(m)\downarrow, \\ \uparrow & \text{otherwise.} \end{cases}$$

By the uniform coding property, find a recursive function g such that $\psi(m, x) = \Phi_{g(m)}(x)$ for all m and x. Given a recursive function f, let e be a Gödel number for fg. Show that $n = g(e)$ is a fixed point for f.)

Chapter II
Embeddings and Extensions of Embeddings in the Degrees

We define the degrees of unsolvability in this chapter, and show that these degrees from an uppersemilattice. Much of the rest of this book will be devoted to studying this uppersemilattice. The study begins in this chapter, with sections on embedding theorems and on extensions of embeddings into the degrees. We also prove the decidability of a certain natural class of sentences about the degrees.

1. Uppersemilattice Structure for the Degrees

We are now ready to define the degrees of unsolvability, and to show that Turing reducibility induces a partial ordering on these degrees which gives rise to an uppersemilattice. In Section 4 we will prove that the degrees do not form a lattice.

We begin with some algebraic definitions.

1.1. Definition. A *partially ordered set* (*poset*) $\langle P, \leqslant \rangle$ is a set P together with a binary relation $\leqslant \subseteq P^2$ having the following properties:

(i) *Reflexivity*: $\forall x \in P(x \leqslant x)$.
(ii) *Antisymmetry*: $\forall x, y \in P(x \leqslant y \,\&\, y \leqslant x \rightarrow x = y)$.
(iii) *Transitivity*: $\forall x, y, z \in P(x \leqslant y \,\&\, y \leqslant z \rightarrow x \leqslant z)$.

1.2 Definition. An *uppersemilattice* (*usl*) is a triple $\langle P, \leqslant, \vee \rangle$ such that $\langle P, \leqslant \rangle$ is a poset, and $\vee : P^2 \rightarrow P$ (write $x \vee y = z$ for $\vee(x, y) = z$) satisfies:

(i) $\qquad \forall x, y \in P(x \leqslant x \vee y \,\&\, y \leqslant x \vee y)$

and

(ii) $\qquad \forall x, y, u \in P(x \leqslant u \,\&\, y \leqslant u \rightarrow x \vee y \leqslant u)$.

Thus a usl is a poset in which every pair of elements has a least upper bound.

Clause (ii) of Definition 1.1 prevents the use of $\leqslant_T$ to directly transform N^N into a poset. This obstruction is circumvented by using certain equivalence classes of N^N, *the degrees*, as the domain of the poset. The equivalence relation used is the following.

1.3 Definition. For $f, g \in N^N$, define $f \equiv_T g$ if $f \leqslant_T g$ and $g \leqslant_T f$.

We leave the proof of the fact that $\equiv_T$ is an equivalence relation to the reader (Exercises 1.11 and 1.12). $\equiv_T$ partitions N^N into equivalence classes which are now defined.

1.4 Definition. Let $f \in N^N$ be given. The *degree* (*of unsolvability*) *of* f, $\mathbf{f}$, is $\{g \in N^N: g \equiv_T f\}$.

1.5 Notation. $\{\mathbf{f}: f \in N^N\}$ will henceforth be denoted by $\mathbf{D}$.

1.6 Remark. Since $|N^N| = 2^{\aleph_0}$ and for each $\mathbf{d} \in \mathbf{D}$, $|\mathbf{d}| = \aleph_0$, a simple computation in cardinal arithmetic shows that $|\mathbf{D}| = 2^{\aleph_0}$.

The next two definitions indicate the natural way in which usl structure is induced on $\mathbf{D}$.

1.7 Definition. Let $\mathbf{a}, \mathbf{b} \in \mathbf{D}$ be given. We say that $\mathbf{a} \leqslant \mathbf{b}$ if

$$\forall f, g \in N^N (f \in \mathbf{a} \,\&\, g \in \mathbf{b} \rightarrow f \leqslant_T g).$$

We leave it to the reader (Exercise 1.13) to show that

$$\mathbf{a} \leqslant \mathbf{b} \Leftrightarrow \exists f, g \in N^N (f \in \mathbf{a} \,\&\, g \in \mathbf{b} \,\&\, f \leqslant_T g).$$

1.8 Definition. Let $\mathbf{a}, \mathbf{b} \in \mathbf{D}$, $f \in \mathbf{a}$ and $g \in \mathbf{b}$ be given. Define $\mathbf{a} \cup \mathbf{b}$ to be the degree of the function $f \oplus g \in N^N$ defined by

$$f \oplus g(x) = \begin{cases} f(x/2) & \text{if } x \text{ is even,} \\ g(x - 1/2) & \text{if } x \text{ is odd.} \end{cases}$$

Let $\mathscr{D} = \langle \mathbf{D}, \leqslant \rangle$ and $\mathscr{D}\mathscr{U} = \langle \mathbf{D}, \leqslant, \cup \rangle$. We leave it to the reader (Exercise 1.14) to verify that $\mathscr{D}$ is a poset and that $\mathscr{D}\mathscr{U}$ is a usl. Note that $\mathbf{D}$ has a smallest element, namely, the degree of the recursive functions (Exercise 1.16).

1.9 Notation. We will write $\mathbf{a} = \mathbf{b}$ for $\mathbf{a} \leqslant \mathbf{b}$ and $\mathbf{b} \leqslant \mathbf{a}$. $<, \geqslant, >, \neq$, etc. will have the obvious meaning. $\mathbf{0}$ will denote the smallest degree. $\cup\{\mathbf{a}_i: 1 \leqslant i \leqslant n\}$ will denote $\mathbf{a}_1 \cup \cdots \cup \mathbf{a}_n$, and $\cap\{\mathbf{a}_i: 1 \leqslant i \leqslant n\}$ will denote the greatest element $\mathbf{d} \in \mathbf{D}$ such that $\mathbf{d} \leqslant \mathbf{a}_i$ for $i = 1, 2, \ldots, n$ if such an element exists, and will be undefined otherwise.

The study of relative recursion, or equivalently, computation from oracles leads naturally to the study of the degrees. Questions about information contained in functions which can be computed from an f oracle are best formulated in terms of the structure of the degrees below f. Hence the study of $\mathscr{D}$ will shed light on relative recursion.

Several algebraic and logical problems arise naturally in the study of $\mathscr{D}$. We would like to have a classification of the usls which can be embedded into $\mathscr{D}$, and to develop structure theory for $\mathscr{D}$. We would like to have answers to certain questions about the elementary theory of $\mathscr{D}$, e.g., "is the theory decidable?", and "how complicated is this theory?". Some of these questions have been answered, while a complete answer to the others still remains to be found. (Note that for the questions mentioned above, $\mathscr{D}$ and $\mathscr{D}\mathscr{U}$ are interchangeable.) These, and other questions will be studied in this book, a study which begins in the next section.

1.10 Remark. $\mathscr{D}$ was first defined and studied by Kleene and Post [1954]. This paper has an interesting history. Kleene received a letter from Post with some of the definitions and theorems, and suggested that Post publish those results. Post was reluctant to do so, feeling that some of the most important initial questions about the degrees had not yet been answered. Some of these questions were later answered by Kleene, who added his results to Post's and had the paper published. This was done while Post was terminally ill, and we do not know whether or not Post ever read the paper.

1.11–1.17 Exercises

***1.11** Show that $\leqslant_T$ is transitive.

***1.12** Show that $\equiv_T$ is an equivalence relation.

***1.13** Show that $\mathbf{a} \leqslant \mathbf{b} \Leftrightarrow \exists f, g \in N^N (f \in \mathbf{a} \mathbin{\&} g \in \mathbf{b} \mathbin{\&} f \leqslant_T g)$.

***1.14** Show that $\mathscr{D}\mathscr{U}$ is a usl.

***1.15** Show that every degree contains a set (i.e., a characteristic function).

***1.16** Show that for all degrees $\mathbf{a}, \mathbf{0} \leqslant \mathbf{a}$.

***1.17** Given $\{f_i\colon N \to N\colon i = 0, 1, \ldots, n-1\}$, define $\oplus_{i=0}^{n-1} f_i\colon N \to N$ by $(\oplus_{i=0}^{n-1} f_i)(nx + b) = f_b(x)$ where $0 \leqslant b < n$. Show that $\oplus_{i=0}^{n-1} f_i$ and $((\cdots((f_0 \oplus f_1) \oplus f_2) \oplus \cdots) \oplus f_{n-1})$ have the same degree.

2. Incomparable Degrees

Embeddings into the degrees are considered in this section. Many constructions of classes of degrees with various properties can be carried out through the use of the method of forcing. We describe forcing in this section, and use it to construct incomparable degrees.

Rather than begin immediately with the abstract notion of forcing, we first give a classical proof of the existence of incomparable degrees. We next describe the relationship between this proof and the forcing proof. Forcing is then introduced, and is used to prove the same theorem.

2.1 Definition. Let $\mathbf{a}, \mathbf{b} \in \mathbf{D}$ be given. Then $\mathbf{a}$ and $\mathbf{b}$ are *incomparable* (write $\mathbf{a} \mid \mathbf{b}$) if $\mathbf{a} \not\leqslant \mathbf{b}$ and $\mathbf{b} \not\leqslant \mathbf{a}$.

2.2 Theorem. *There exist* $\mathbf{a}_0, \mathbf{a}_1 \in \mathbf{D}$ *such that* $\mathbf{a}_0 \mid \mathbf{a}_1$.

Proof. We construct sets $A_0, A_1 \subseteq N$ such that $A_0 \not\leqslant_T A_1, A_1 \not\leqslant_T A_0$ and set $\mathbf{a_i} = \mathbf{A_i}$ for $i = 0, 1$. By the Enumeration Theorem, it suffices to satisfy the *requirements*

$$(1) \qquad P_{e,i}\colon \Phi_e^{A_i} \neq A_{1-i}$$