J. Donald Monk

Mathematical Logic

Springer-Verlag

New York Heidelberg Berlin

1976

J. Donald Monk
Department of Mathematics
University of Colorado
Boulder, Colorado 80302

AMS Subject Classifications
Primary: 02-xx
Secondary: 10N-xx, 06-XX, 08-XX, 26A98

Library of Congress Cataloging in Publication Data

Monk, James Donald, 1930–
Mathematical logic.
(Graduate texts in mathematics ; 37)
Bibliography
Includes indexes.
1. Logic, Symbolic and mathematical. I. Title. II. Series.
QA9.M68 511'.3 75–42416

Printed in the United States of America

ISBN 0–387–90170–1 Springer-Verlag New York

ISBN 3–540–90170–1 Springer-Verlag Berlin Heidelberg

to Dorothy

Preface

This book is a development of lectures given by the author numerous times at the University of Colorado, and once at the University of California, Berkeley. A large portion was written while the author worked at the Forschungsinstitut für Mathematik, Eidgennössische Technische Hochschule, Zürich.

A detailed description of the contents of the book, notational conventions, etc., is found at the end of the introduction.

The author's main professional debt is to Alfred Tarski, from whom he learned logic. Several former students have urged the author to publish such a book as this; for such encouragement I am especially indebted to Ralph McKenzie.

I wish to thank James Fickett and Stephen Comer for invaluable help in finding (some of the) errors in the manuscript. Comer also suggested several of the exercises.

J. Donald Monk
October, 1975

Contents

Part III

Decidable and Undecidable Theories 231

Part IV

Model Theory 309

Part V

Unusual Logics 471

Introduction

Leafing through almost any exposition of modern mathematical logic, including this book, one will note the highly technical and purely mathematical nature of most of the material. Generally speaking this may seem strange to the novice, who pictures logic as forming the foundation of mathematics and expects to find many difficult discussions concerning the philosophy of mathematics. Even more puzzling to such a person is the fact that most works on logic presuppose a substantial amount of mathematical background, in fact, usually more set theory than is required for other mathematical subjects at a comparable level. To the novice it would seem more appropriate to begin by assuming nothing more than a general cultural background. In this introduction we want to try to justify the approach used in this book and similar ones. Inevitably this will require a discussion of the philosophy of mathematics. We cannot do full justice to this topic here, and the interested reader will have to study further, for example in the references given at the end of this introduction. We should emphasize at the outset that the various possible philosophical viewpoints concerning the nature or purpose of mathematics do not effect one way or the other the correctness of mathematical reasoning (including the technical results of this book). They do effect how mathematical results are to be intuitively interpreted, and which mathematical problems are considered as more significant.

We shall discuss first a possible definition of mathematics, and then turn to a deeper discussion of the meaning of mathematics. After this we can in part justify the methods of modern logic described in this book. The introduction closes with an outline of the contents of the book and some comments on notation.

As a tentative definition of mathematics, we may say it is an *a priori*, *exact*, *abstract*, *absolute*, *applicable*, and *symbolic* scientific discipline. We now consider these defining characteristics one by one. To say that mathematics is *a priori* is to say that it is independent of experience. Unlike physics or chemistry, the laws of mathematics are not laws of nature or dependent upon laws of nature. Theorems would remain valid in other possible worlds, where the laws of physics might be entirely different. If we take mathematical knowledge to mean a body of theorems and their formal proofs, then we can say that such knowledge is independent of all experience except the very rudimentary process of mechanically checking that the proofs are really proofs in the logical sense—lists of formulas subject to rules of inference. Of course this is a very limited conception of mathematical knowledge, but there can be little doubt that, so conceived, it is *a priori* knowledge. Depending on one's attitude towards mathematical truth, one might wish to broaden this view of mathematical knowledge; we shall discuss this later. Under broadened views, it is certainly possible to challenge the *a priori* nature of mathematics; see, e.g., Kalmar [6] (bibliography at the end of this introduction).

Mathematics is *exact* in the sense that all its terms, definitions, rules of proof, etc. have a precise meaning. This is especially true when mathematics is based upon logic and set theory, as it is customary to do these days. This aspect of mathematics is perhaps the main thing that distinguishes it from other scientific disciplines. The possibility of being exact stems partially from its *a priori* nature. It is of course difficult to be very precise in discussing empirical evidence, because nature is so complex, difficult to classify, observations are subject to experimental error, etc. But in the realm of ideas divorced from experience it is possible to be precise, and in mathematics one is precise. Of course some parts of philosophical speculation are concerned with *a priori* matters also, but such speculation differs from mathematics in not being exact.

Another distinguishing feature of mathematical discourse is that it is generally much more *abstract* than ordinary language. One of the hallmarks of modern mathematics is its abstractness, but even classical mathematics is very abstract compared to other disciplines. Number, line, plane, etc. are not concrete concepts compared to chairs, cars, or planets. There are different levels of abstractness in mathematics, too; one may contemplate a progression like numbers, groups, universal algebras, categories. This characteristic of mathematics is shared by many other disciplines. In physics, for example, discussion may range from very concrete engineering problems to possible models for atomic nuclei. But in mathematics the concepts are *a priori*, already implying some degree of abstractness, and the tendency toward abstractness is very rampant.

Next, mathematical results are *absolute*, not revisable on the basis of experience. Again, viewing mathematics just as a collection of theorems and formal proofs, there is little to quarrel with in this statement. Thus we see once more a difference between mathematics and experimental evidence; the latter is certainly subject to revision as measurements become more exact.

Of course the appropriateness of a mathematical discipline for a given empirical study is highly subject to revision. Experimental evidence and *a posteriori* reasoning hence play a role in motivation for studying parts of mathematics and in the directions for mathematical research. One's attitude toward the absoluteness of mathematics is also colored by differing commitments to the nature of mathematical truth (see below).

A feature of mathematics which is probably not inherent in its nature is its *applicability*. A very great portion of mathematics arises by trying to give a precise mathematical theory for some concrete, perhaps even nonmathematical, situation. Of course geometry and much of classical mathematics arose in this way from special intuition derived from actual sense evidence. Also, logic owes much to this means of development; formal languages arose from less formal mathematical discourse, the notion of Turing machine from the intuitive notion of computability, etc. Many very abstract mathematical disciplines arose from an analysis of less abstract parts of mathematics, and may hence be subsumed under this facet of the discipline; group theory and algebraic topology may be mentioned as examples. This aspect of mathematics is emphasized in Rogers [12], for example.

Finally, the use of *symbolic* notation is a main characteristic of mathematics. This is connected with its exact nature, but even more connected with the development of mathematics as a kind of language. In fact, mathematics is often said just to be a language of a special kind. Most linguists would reject this claim, for mathematics fails to satisfy many of their criteria for a language, e.g., that of universality (capability of expressing usual events, emotions, ideas, etc. which occur in ordinary life). But mathematics does have many features in common with ordinary languages. It has proper names, such as π and e, and many mathematical statements have a subject–predicate form. In fact, almost all mathematical statements can be given an entirely nonsymbolic rendering, although this may be awkward in many cases. Thus mathematics can be considered as embedded in the particular natural language—English, Russian, etc.—in which it is partially expressed. But also mathematics can, in principle, be expressed purely symbolically; indeed, a large portion of mathematics was so expressed in Russell and Whitehead's *Principia Mathematica*.

Now we turn to a discussion of the nature of mathematical truth. We shall briefly mention three opposed views here: platonism, formalism, and intuitionism. The views of most mathematicians as to what their subject is all about are combinations of these three. On a subjective evaluation, we would estimate the mathematical world as populated with 65% platonists, 30% formalists, and 5% intuitionists. We describe here the three extremes. There are (perhaps) more palatable versions of all three.

According to extreme *platonism*, mathematical objects are real, as real as any things in the world we live in. For example, infinite sets exist, not just as a mental construct, but in a real sense, perhaps in a "hyperworld." Similarly, nondenumerable sets, real numbers, choice functions, Lebesgue measure, and the category of all categories have a real existence. Since all of the mathematical objects are real, the job of a mathematician is as empirical as that of a geologist or physicist; the mathematician looks at a special aspect of nature and tries to discover some of the facts. The various mathematical statements, like the Riemann hypothesis or the continuum hypothesis, are either true or false in the real world. The axioms of set theory are axioms in the Greek sense—self-evident statements which form a partial basis to deductively arrive at other truths. Hence such results as the independence of the continuum hypothesis relative to the usual set-theoretical axioms force the platonist into a search for new insights and intuitions into the nature of sets so as to decide the truth or falsity of those statements which cannot be decided upon the basis of already accepted facts. Thus for him the independence results are not results about mathematics, but just about the formalization of mathematics. This view of mathematics leads to some revisions of the "definition" of mathematics we gave earlier. Thus it no longer is independent of empirical facts, but is as empirical as physics or chemistry. But since a platonist will still insist upon the absolute, immutable nature of mathematics, it still has an *a priori* aspect. For more detailed accounts of platonism see Mostowski [10] or Gödel [3].

In giving the definition of mathematics we have implicitly followed the view of formalists. A *formalist* does not believe that any mathematical objects have a real existence. For him, mathematics is just a collection of axioms, theorems, and formal proofs. Of course, the activity of mathematics is not just randomly writing down formal proofs for random theorems. The choices of axioms, of problems, of research directions, are influenced by a variety of considerations—practical, artistic, mystical—but all really nonmathematical. A revised version of platonism is to think of mathematical concepts not as actually existing but as mental constructs. A very extensive understructure for much of formalism is very close to this version of platonism—the formal development of a mathematical theory to correspond to certain mental constructions. Good examples are geometry and set theory, both of which have developed in this way. And all concept analysis (e.g., analyzing the intuitive notion of computability) can be viewed as philosophical bases for much formal mathematics. Another motivating principle behind much formalism is the desire to inter-relate different parts of mathematics; for example, one may cite the ties among sentential logic, Boolean algebra, and topology. Thus while mathematics itself is precise and formal, a mathematician is more of an artist than an experimental scientist. For more on formalism, see Hilbert [5], A. Robinson [11], and P. Cohen [2]. For another discussion of platonism and formalism see Monk [9].

Intuitionism is connected with the constructivist trend in mathematics: a mathematical object exists only if there is a (mental) construction for it. This philosophy implies that much ordinary mathematics must be thrown out, while platonism and formalism can both be used to justify present day mathematics. Even logical principles themselves must be modified on the basis of intuitionism. Thus the law of excluded middle—for any statement A, either A holds or (*not* A) holds—is rejected. The reasoning here goes as follows. Let A, for example, be the statement that there are infinitely many primes p such that $p + 2$ is also a prime. Then A does not presently hold, for we do not possess a construction which can go from any integer m given to us and produce primes p and $p + 2$ with $m < p$. But (*not* A) also does not hold, since we do not possess a construction which can go from any hypothetical construction proving A and produce a contradiction. One may say that intuitionism is the only branch of mathematics dealing directly with real, constructible objects. Other parts of mathematics introduce idealized concepts which have no constructive counterpart. For most mathematicians this idealism is fully justified, since one can make contact with verifiable, applicable mathematics as an offshoot of idealistic mathematics. See Heyting [4] and Bishop [1].

Now from the point of view of these brief comments on the nature of mathematics let us return to the problem of justifying our purely technical approach to logic. First of all, we do want to consider logic as a branch of mathematics, and subject this branch to as severe and searching an analysis as other branches. It is natural, from this point of view, to take a no-holds-barred attitude. For this reason, we shall base our discussion on a set-theoretical foundation like that used in developing analysis, or algebra, or topology. We may consider our task as that of giving a mathematical analysis of the basic concepts of logic and mathematics themselves. Thus we treat mathematical and logical practice as given empirical data and attempt to develop a purely mathematical theory of logic abstracted from these data. Our degree of success, that is, the extent to which this abstraction corresponds to the reality of mathematical practice, is a matter for philosophers to discuss. It will be evident also that many of our technical results have important implications in the philosophy of mathematics, but we shall not discuss these. We shall make some comments concerning an application of technical logic within mathematics, namely to the precise development of mathematics. Indeed, mathematics, formally developed, starts with logic, proceeds to set theory, and then branches into its several disciplines. We are not in the main concerned with this development, but a proper procedure for such a development will be easy to infer from the easier portions of our discussion in this book.

Inherent in our treatment of logic, then, is the fact that our whole discussion takes place within ordinary intuitive mathematics. Naturally, we do not develop this intuitive mathematics formally here. Essentially all that we presuppose is elementary set theory, such as it is developed in Monk [8] for example. (See the end of this introduction for a description of set-theoretic notation we use that is not standard.) Since our main concern in the book is certain formal languages, we thus are confronted with two levels of language in the book: the informal metalanguage, in which the whole discussion takes place, and the object languages which we discuss. The latter will be defined, in due course, as certain sets (!), in keeping with the foundation of all mathematics upon set theory. It is important to keep sharply in mind this distinction between language and metalanguage. But it should also be emphasized that many times we take ordinary metalanguage arguments and "translate" them into a given formal language; see Chapter 17, for example.

Briefly speaking, the book is divided up as follows. Part I is devoted to the elements of recursive function theory—the mathematical theory of effective, machine-like processes. The most important things in Part I are the various equivalent definitions of recursive functions. In Part II we give a short course in elementary logic, covering topics frequently found in undergraduate courses in mathematical logic. The main results are the completeness and compactness theorems. The heart of the book is in the remaining three parts. Part III treats one of the two basic questions of mathematical logic: given a theory T, is there an automatic method for determining the validity of sentences in T? Aside from general results, the chapter treats this question for many ordinary theories, with both positive and negative results. For example, there is no such method for set theory, but there is for ordinary addition of integers. As corollaries we present celebrated results of Gödel concerning the incompleteness of strong theories and the virtual impossibility of giving convincing consistency proofs for strong theories. The second basic question of logic is treated in Part IV: what is the relationship between semantic properties of languages (truth of sentences, denotations of words, etc.) and formal characteristics of them (form of sentences, etc.)? Some important results of this chapter are Beth's completeness theorem for definitions, Lindstrom's abstract characterization of languages, and the Keisler–Shelah mathematical characterization of the formal definability of classes of structures. In both of these chapters the languages studied are of a comprehensive type known as first-order languages. Other popular languages are studied in Part V, e.g., the type theory first extensively developed by Russell and Whitehead and the languages with infinitely long expressions.

Optional chapters in the book are marked with an asterisk *. For the interdependence of the chapters, see the graph following this introduction. The book is provided with approximately 320 exercises. Difficult or lengthy ones are marked with an asterisk *. Most of the exercises are not necessary for further work in the book; those that are are marked with a prime ′. The end of a proof is signaled by the symbol □.

As already mentioned, we will be following the set-theoretical notation found in [8]. For the convenience of the reader we set out here the notation from [8] that is not in general use. For informal logic we use "$\Rightarrow$" for "implies," "$\Leftrightarrow$" or "iff" for "if and only if," "$\neg$" for "not," "$\forall$" for "for all," and "$\exists$" for "there exists." We distinguish between classes and sets in the usual fashion. The notation $\{x : \varphi(x)\}$ denotes the class of all sets x such that $\varphi(x)$. Inclusion and proper inclusion are denoted by $\subseteq$ and $\subset$ respectively. The empty set is denoted by 0, and is the same as the ordinal number 0. We let $A \sim B = \{x : x \in A, x \notin B\}$. The ordered pair (a, b) is defined by $(a, b) = \{\{a\}, \{a, b\}\}$; and $(a, b, c) = ((a, b), c)$, $(a, b, c, d) = ((a, b, c), d)$, etc. A binary relation is a set of ordered pairs; ternary, quaternary relations are defined similarly. The domain and range of a binary relation R are denoted by Dmn R and Rng R respectively. *The value of a function f at an argument a is denoted variously by ${}^a f$, ${}_a f$, f^a, f_a, fa, $f(a)$; and we may change notation frequently, especially for typographical reasons.* The symbol $\langle \tau(i) : i \in I \rangle$ denotes a function f with domain I such that $fi = \tau(i)$ for all $i \in I$. The sequence $\langle x_0, \ldots, x_{m-1} \rangle$ is the function with domain m and value x_i for each $i \in m$. The set ${}^A B$ is the set of all functions mapping A into B. An m-ary relation is a subset of ${}^m A$, for some A. Thus a 2-ary relation is a set of ordered pairs, $\langle x, y \rangle$. By abuse of notation we shall sometimes identify the two kinds of ordered pairs, of binary relations, ternary relations, etc. We write f^*A for $\{fa : a \in A\}$. The notations $f\colon A \to B$, $f\colon A \twoheadrightarrow B$, $f\colon A \rightarrowtail B$, and $f\colon A \rightarrowtail\!\!\!\!\twoheadrightarrow B$ mean that f is a function mapping A into (onto, one-one into, one-one onto respectively) B. The identity function (on the class of all sets) is denoted by $\boldsymbol{I}$. The restriction of a function F to a set A is denoted by $F \upharpoonright A$. The class of all subsets of A is denoted by $\boldsymbol{S}A$. Given an equivalence relation R on a set A, the equivalence class of $a \in A$ is denoted by $[a]_R$ or $[a]$, while the set of all equivalence classes is denoted by A/R. Ordinals are denoted by small Greek letters $\alpha, \beta, \gamma, \ldots$, while cardinals are denoted by small German letters $\mathfrak{m}, \mathfrak{n}, \ldots$. The cardinality of a set A is denoted by $|A|$. The least cardinal greater than a cardinal $\mathfrak{m}$ is denoted by $\mathfrak{m}^+$. *For typographical reasons we sometimes write* (exp $(\mathfrak{m}, \mathfrak{n})$ *for* $\mathfrak{m}^{\mathfrak{n}}$ *and* exp $\mathfrak{m}$ *for* $2^{\mathfrak{m}}$.

One final remark on our notation throughout the book: *in various symbolisms introduced with superscripts or subscripts, we will omit the latter when no confusion is likely* (e.g., $[a]_R$ and $[a]$ above).

BIBLIOGRAPHY

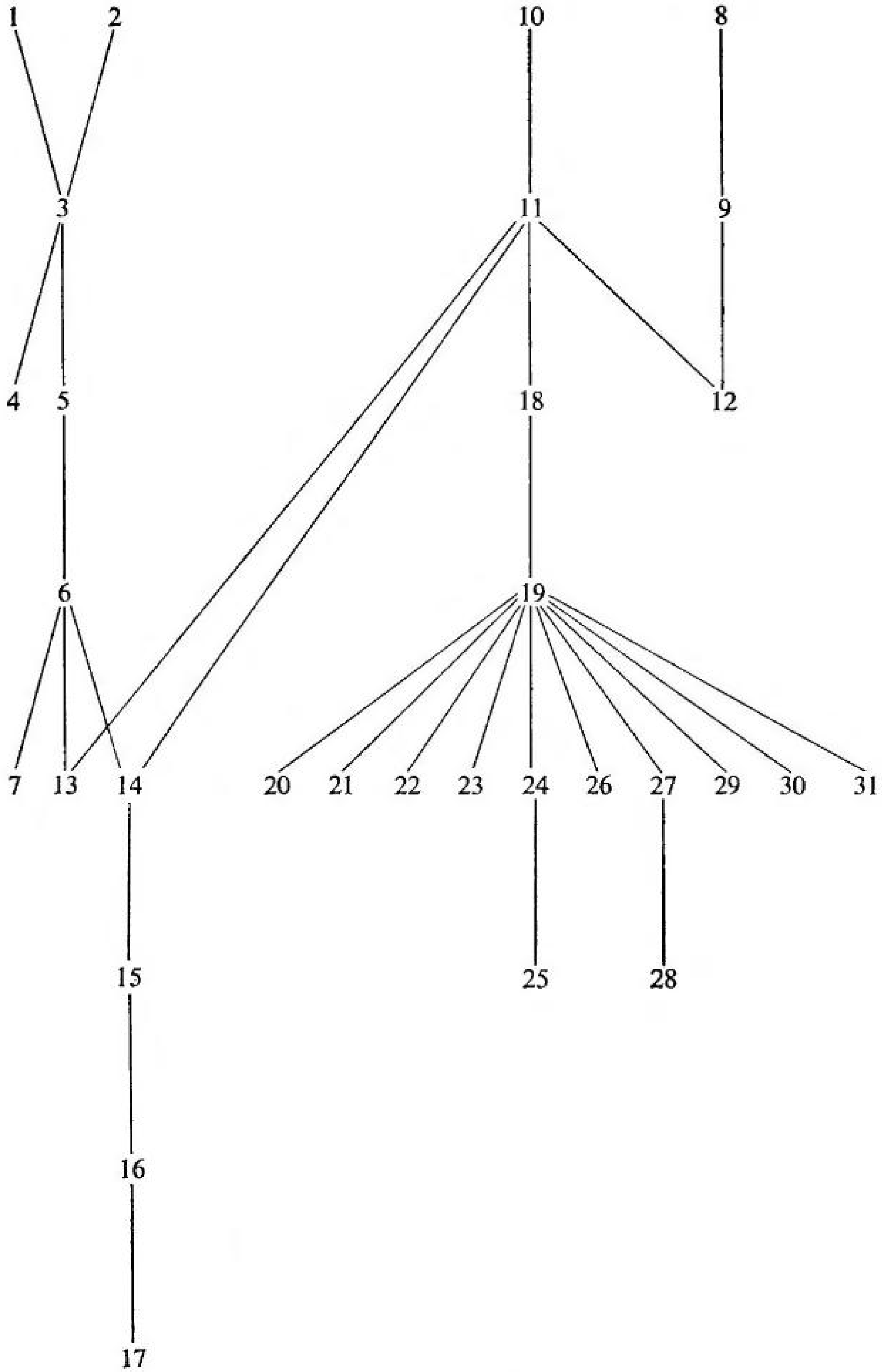
1. Bishop, E. *Foundations of Constructive Analysis.* New York: McGraw-Hill (1967).
2. Cohen, P. Comments on the foundations of set theory. In: *Axiomatic Set Theory*. Providence: Amer. Math. Soc. (1971), 9–16.
3. Gödel, K. What is Cantor's continuum problem? *Amer. Math. Monthly*, **54** (1947), 515–525.
4. Heyting, A. *Intuitionism.* Amsterdam: North-Holland (1966).
5. Hilbert, D. Die logischen Grundlagen der mathematik. *Math. Ann.*, *88* (1923), 151–165.
6. Kalmar, L. Foundations of mathematics—whither now. In: *Problems in the Philosophy of Mathematics.* Amsterdam: North-Holland (1967), 187–194.
7. Kreisel, G. Observations on popular discussions of foundations. In: *Axiomatic Set Theory.* Providence: Amer. Math. Soc. (1971), 189–198.
8. Monk, J. D. *Introduction to Set Theory.* New York: McGraw-Hill (1969).

9. Monk, J. D. On the foundations of set theory. *Amer. Math. Monthly*, *77* (1970), 703–711.

10. Mostowski, A. Recent results in set theory. In: *Problems in the Philosophy of Mathematics*. Amsterdam: North-Holland (1967), 82–96.

11. Robinson, A. Formalism 64. In: *Logic, Methodology, and the Philosophy of Science*. Amsterdam: North-Holland (1964), 228–246.

12. Rogers, R. Mathematical and philosophical analyses. *Philos. Sci.*, *31* (1964), 255–264.

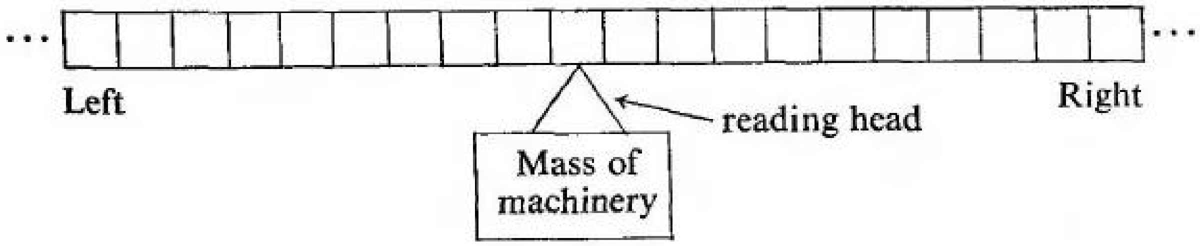


Interdependence of Chapters

1 Turing Machines

In this chapter we shall present a popular mathematical version of effectiveness, Turing computability, which will form our main rigorous basis for the mathematical discussion of effectivity. Actually in this section we present only some of the basic definitions concerning Turing machines and some elementary results which both illuminate these definitions and form a basis for later work. The definition of Turing computability itself is found in Chapter 3. After giving the formal definition of a Turing machine we discuss briefly the motivation behind the definition.

In our exposition of Turing machines we follow Hermes [2] rather closely. A Turing machine (intuitively) consists of a mass of machinery, a reading head, and a tape infinite in both directions. The machine may be in any of finitely many *internal states*. The tape is divided up into squares called *fields* of the tape (see figure).



The machine proceeds step by step. At a given step it takes an action depending on what state it is in and upon what it finds on the field that the reading head is on. We allow only two symbols, 0 and 1, to be on a given field, and all but finitely many of the fields have 0 on them. These are the actions the machine can take:

(1) Write 0 on the given field (first erasing what is there).
(2) Write 1 on the given field (first erasing what is there).

(3) Move tape one square to the right.
(4) Move tape one square to the left.
(5) Stop.

We now want to make this rigorous.

Definition 1.1. A *Turing machine* is a matrix of the form

$$\begin{array}{cccc}
c_1 & 0 & v_1 & d_1 \\
c_1 & 1 & v_2 & d_2 \\
c_2 & 0 & v_3 & d_3 \\
c_2 & 1 & v_4 & d_4 \\
\vdots & \vdots & \vdots & \vdots \\
c_n & 0 & v_{2n-1} & d_{2n-1} \\
c_n & 1 & v_{2n} & d_{2n} \\
\vdots & \vdots & \vdots & \vdots \\
c_m & 0 & v_{2m-1} & d_{2m-1} \\
c_m & 1 & v_{2m} & d_{2m}
\end{array}$$

where: $c_1, \ldots, c_m$ are distinct members of ω, $v_1, \ldots, v_{2m} \in \{0, 1, 2, 3, 4\}$ and $d_1, \ldots, d_{2m} \in \{c_1, \ldots, c_m\}$. $c_1, \ldots, c_m$ are called *states*. c_1 is called the *initial state* of the machine.

We think of a row $c_i\ \varepsilon\ v_j\ d_j$ of this matrix as giving the following information: when the machine is in state c_i and scans the symbol ε on the tape, it takes action v_j and then moves into state d_j. Here the action given by v_j is as follows:

$v_j = 0$: write 0 on scanned square;
$v_j = 1$: write 1 on scanned square;
$v_j = 2$: move tape one square to the right;
$v_j = 3$: move tape one square to the left;
$v_j = 4$: stop.

To make *this* precise, we proceed as follows:

Definition 1.2. Let $\mathbb{Z}$ be the set of all (negative and nonnegative) integers. A *tape description* is a function F mapping $\mathbb{Z}$ into $\{0, 1\}$ which is 0 except for finitely many values. A *configuration* of a given Turing machine T is a triple (F, d, e) such that F is a tape description, d is a state, and e is an integer (which tells us, intuitively, where the reading head is). A *computation step* of T is a pair $((F, d, e), (F', d', e'))$ of configurations such that: if the line of the Turing machine beginning with (d, Fe) is (d, Fe, w, f), then:

$$\begin{array}{llll} \text{if } w = 0 & \text{then } F' = F_0^e, & d' = f, & e' = e; \\ \text{if } w = 1 & \text{then } F' = F_1^e, & d' = f, & e' = e; \\ \text{if } w = 2 & \text{then } F' = F, & d' = f, & e' = e - 1; \\ \text{if } w = 3 & \text{then } F' = F, & d' = f, & e' = e + 1. \end{array}$$

Here F_ε^e is the function $(F \sim \{(e, Fe)\}) \cup \{(e, \varepsilon)\}$. Thus F_ε^e is the tape description acting like F except possibly at e, and $F_\varepsilon^e e = \varepsilon$. A *computation* of T is a finite sequence $\langle (F_0, d_0, e_0), \ldots, (F_m, d_m, e_m) \rangle$ of configurations such that $d_0 = c_1$, $((F_i, d_i, e_i), (F_{i+1}, d_{i+1}, e_{i+1}))$ is a computation step for each $i < m$, and the row of the Turing machine beginning (d_m, Fe_m) has 4 as its third entry.

The way a Turing machine runs has now been described. To compute a function f, roughly speaking we hand the machine a number x and it produces fx as an output. Since only zeros and ones appear on a tape, we cannot literally hand x to the machine; it must be coded by zeros and ones. The mathematically most obvious way of coding x is to use its binary representation as a "decimal" with base 2. However, this is inconvenient, in view of the very primitive operations which a Turing machine can perform. We elect instead to represent x by a sequence of $x + 1$ one's. (This is sometimes called the *tally* notation.) The extra "one" is added in order to be able to recognize the code of the number zero as different from a zero entry on the tape whose purpose is just as a blank. The precise way in which functions are computed by a Turing machine will be defined in Chapter 3. In this chapter we want to see how these rather primitive looking machines can nevertheless perform some intricate operations on strings of zeros and ones. These results will be useful in Chapter 3 and later work.

Using the intuitive notion of coding we can argue as follows that Turing machines are really quite powerful: We have seen informally how to represent any number on a tape. A sequence of numbers can be represented by putting blanks (zeros) between the strings of ones representing the numbers. By using two blanks one can code several blocks of numbers, or one can use the two blanks to recognize a portion of the tape set aside for a special purpose. By repeated adjoining of a one, it is possible to add with a Turing machine; and by repeated addition, one can multiply. Since a new state depends on the currently scanned symbol, it is possible to set up different actions depending upon what is on the tape. And we are not really restricted to just one square in this decision making, since by using several states we can examine any restricted portion of the tape.

In the general theory of Turing machines, one allows several symbols instead of just 0 and 1 (see, e.g., [2]). However, it is clearly possible to code these different symbols by different strings of 1's. Several tapes may also be allowed. Again such a modification can be coded within our machines; in the case of two tapes, for example, one may instead use odd and even numbered squares on a single tape.

These intuitive comments on the strength of Turing machines of course would require proof. Some of them will be proved later, and we hope that they will all seem plausible after we have worked with Turing machines a while. For a more detailed argument on the strength of Turing machines see the introduction to [2].

Definition 1.3. T_{right} is the following machine:

$$\begin{matrix} 0 & 0 & 2 & 1 \\ 0 & 1 & 2 & 1 \\ 1 & 0 & 4 & 1 \\ 1 & 1 & 4 & 1 \end{matrix}$$

Proposition 1.4. *For any tape description F and any $e \in \mathbb{Z}$, $\langle (F, 0, e), (F, 1, e-1)\rangle$ is a computation of T_{right}.*

Thus T_{right} merely moves the tape one square to the right, and then stops.

Definition 1.5. T_{left} is the following machine:

$$\begin{matrix} 0 & 0 & 3 & 1 \\ 0 & 1 & 3 & 1 \\ 1 & 0 & 4 & 1 \\ 1 & 1 & 4 & 1 \end{matrix}$$

Proposition 1.6. *For any tape description F and any $e \in \mathbb{Z}$, $\langle (F, 0, e), (F, 1, e+1)\rangle$ is a computation of T_{left}.*

Thus T_{left} moves the tape one square to the left and then stops.

Definition 1.7. T_0 is the following machine:

$$\begin{matrix} 0 & 0 & 4 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

Proposition 1.8. *For any tape description F and any $e \in \mathbb{Z}$, (i) if $Fe = 0$, then $\langle (F, 0, e)\rangle$ is a computation of T_0; (ii) if $Fe = 1$, then $\langle (F, 0, e), (F_0^e, 0, e)\rangle$ is a computation of T_0. Thus T_0 writes a 0 if a zero is not here, but does not move the tape.*

Definition 1.9. T_1 is the following machine:

$$
\begin{array}{cccc}
0 & 0 & 1 & 0 \\
0 & 1 & 4 & 0
\end{array}
$$

Proposition 1.10. *For any tape description F and any $e \in \mathbb{Z}$, (i) if $Fe = 0$, then $\langle (F, 0, e), (F_1^e, 0, e) \rangle$ is a computation of T_1; (ii) if $Fe = 1$, then $\langle (F, 0, e) \rangle$ is a computation of T_1. T_1 writes a 1 if a 1 is not there, but does not move the tape.*

Definition 1.11. If a is any set and $m \in \omega$, let $a^{(m)}$ be the unique element of ${}^m\{a\}$. Thus $a^{(m)}$ is an m-termed sequence of a's, $a^{(m)} = \langle a, a, \ldots, a \rangle$ (m times). If x and y are finite sequences, say $x = \langle x_0, \ldots, x_{m-1} \rangle$ and $y = \langle y_0, \ldots, y_{n-1} \rangle$, we let $xy = \langle x_0, \ldots, x_{m-1}, y_0, \ldots, y_{n-1} \rangle$. Frequently we write a for $\langle a \rangle$.

Definition 1.12. $T_{l\,\mathrm{seek}\,0}$ is the following machine:

$$
\begin{array}{cccc}
0 & 0 & 2 & 1 \\
0 & 1 & 2 & 1 \\
1 & 0 & 4 & 1 \\
1 & 1 & 1 & 0
\end{array}
$$

A computation with $T_{l\,\mathrm{seek}\,0}$ can be indicated as follows, where we use an obvious notation:

$$
\begin{array}{rcl}
\text{——}\ 0\ 1^{(m)} & a & \text{————} \\
 & \wedge & \\
\text{——}\ 0\ 1^{(m-1)} & 1 & a\ \text{———} \\
 & \wedge & \\
\text{——}\ 0\ 1^{(m-2)} & 1 & 1\ a\ \text{——} \\
 & \wedge & \\
 & \vdots & \\
\text{————}\ 0 & 1 & 1^{(m-1)}\ a\ \text{—} \\
 & \wedge & \\
\text{—————} & 0 & 1^{(m)}\ a\ \text{——} \\
 & \wedge & \\
 & \uparrow & \\
 & \text{Reading head} &
\end{array}
$$

Thus $T_{l\,\mathrm{seek}\,0}$ finds the first 0 to the left of the square it first looks at and stops at that 0. In this and future cases we shall not formulate an exact theorem describing such a fact; we now feel the reader can in principle translate such informal statements as the above into a rigorous form.

Definition 1.13. $T_{r\,\mathrm{seek}\,0}$ is the following machine:

0	0	3	1
0	1	3	1
1	0	4	1
1	1	1	0

$T_{r\,\mathrm{seek}\,0}$ finds the first 0 to the right of the square it first looks at and stops at that 0.

Definition 1.14. $T_{l\,\mathrm{seek}\,0}$ is the following machine:

0	0	2	1
0	1	2	1
1	0	0	0
1	1	4	1

$T_{l\,\mathrm{seek}\,1}$ finds the first 1 to the left of the square it first looks at and stops at that 1. It may be that no such 1 exists; then the machine continues forever, and no computation exists.

Definition 1.15. $T_{r\,\mathrm{seek}\,1}$ is the following machine:

0	0	3	1
0	1	3	1
1	0	0	0
1	1	4	1

$T_{r\,\mathrm{seek}\,1}$ finds the first 1 to the right of the square it first looks at and stops at that 1. But again, it may be that no such 1 exists.

Definition 1.16. Suppose M, N, and P are Turing machines with pairwise disjoint sets of states. By $M \to N$ we mean the machine obtained by writing down N after M, after *first* replacing all rows of M of the forms $(c\ \ 0\ \ 4\ \ d)$ or $(c'\ \ 1\ \ 4\ \ d')$ by the rows $(c,\ \ 0\ \ 0\ \ e)$ or $(c'\ \ 1\ \ 1\ \ e)$ respectively, where e is the initial state of N. By

$$\begin{tikzcd} M \arrow[r, "\text{if } 0"] \arrow[d, "\text{if } 1"'] & N \\ P \end{tikzcd}$$

we mean the machine obtained by writing down M, then N, then P, after first replacing all rows of M of the forms $(c\ \ 0\ \ 4\ \ d)$ or $(c'\ \ 1\ \ 4\ \ d')$ by the rows (c 0 0 e) or (c' 1 1 e') respectively, where e is the initial state of N and e' is the initial state of P.

Obviously we can change the states of a Turing machine by a one-one mapping without effecting what it does to a tape description. Hence we can apply the notation just introduced to machines even if they do not have pairwise disjoint sets of states. Furthermore, the above notation can be combined into large "flow charts" in an obvious way.

Definition 1.17. $T_{\text{seek}\,1}$ is the following machine:

$$\begin{tikzcd}[column sep=small]
\text{Start} \arrow[d] \\
T_{\text{right}} \arrow[r, "\text{if } 0"] \arrow[d, "\text{if } 1"] & T_1 \arrow[r] & T_{\text{left}} \arrow[r, "\text{if } 0"] \arrow[d, "\text{if } 1"] & T_1 \arrow[r] & T_{l\,\text{seek}\,1} \arrow[r] & T_0 \arrow[r] & T_{\text{right}} \arrow[r, "\text{if } 0"] \arrow[d, "\text{if } 1"] & T_1 \arrow[r] & T_{r\,\text{seek}\,1} \arrow[r] & T_0 \arrow[lllllll, bend right] \\
\text{Stop} & & T_{l\,\text{seek}\,1} \arrow[d] & & & & T_{r\,\text{seek}\,1} \arrow[d] \\
& & T_0 \arrow[d] & & & & T_0 \arrow[d] \\
& & T_{r\,\text{seek}\,1} & & & & T_{l\,\text{seek}\,1}
\end{tikzcd}$$

(Here by $T_{\text{right}} \xrightarrow{\text{if } 1}$ Stop we mean that the row (1 1 4 1) of T_{right} is not to be changed.)

This machine just finds a 1 and stops there. It must look both left and right to find such a 1; 1's are written (but later erased) to keep track of how far the search has gone, so that the final tape description is the same as the initial one. If the tape is blank initially the computation continues forever.

Since this is a rather complicated procedure we again indicate in detail a computation using $T_{\text{seek}\,1}$. First we have two trivial cases:

Starting with $1\ \underset{\wedge}{a}$	*Starting with* $0\ \underset{\wedge}{1}$
$1\ \underset{\wedge}{a}$	$0\ \underset{\wedge}{1}$
$\underset{\wedge}{1}\ a$	$\underset{\wedge}{0}\ 1$
	$\underset{\wedge}{1}\ 1$
	$1\ \underset{\wedge}{1}$
	$\underset{\wedge}{1}\ 1$
	$\underset{\wedge}{0}\ 1$
	$0\ \underset{\wedge}{1}$

In the nontrivial case we start with $—\, 0^{(m)}\, \underset{\wedge}{0}\, 0^{(n)}\, —$; $m > 0$:

Start

$$
\begin{array}{r@{\;\;}l}
0^{(m)} & \underset{\wedge}{0}\; 0^{(n)} \\
0^{(m-1)} & \underset{\wedge}{0}\; 0^{(n+1)} \\
0^{(m-1)} & \underset{\wedge}{1}\; 0^{(n+1)} \\
0^{(m-1)}\; 1 & \underset{\wedge}{0}\; 0^{(n)} \\
 & \vdots \\
0^{(m-i)}\; 1\; 0^{(2i-2)} & \underset{\wedge}{0}\; 0^{(n-i+1)} \\
0^{(m-i)}\; 1\; 0^{(2i-2)} & \underset{\wedge}{1}\; 0^{(n-i+1)} \\
0^{(m-i)} & \underset{\wedge}{1}\; 0^{(2i-2)}\; 1\; 0^{(n-i+1)} \\
0^{(m-i)} & \underset{\wedge}{0}\; 0^{(2i-2)}\; 1\; 0^{(n-i+1)} \\
0^{(m-i-1)} & \underset{\wedge}{0}\; 0^{(2i-1)}\; 1\; 0^{(n-i+1)} \\
0^{(m-i-1)} & \underset{\wedge}{1}\; 0^{(2i-1)}\; 1\; 0^{(n-i+1)} \\
0^{(m-i-1)}\; 1\; 0^{(2i-1)} & \underset{\wedge}{1}\; 0^{(n-i+1)} \\
0^{(m-i-1)}\; 1\; 0^{(2i-1)} & \underset{\wedge}{0}\; 0^{(n-i+1)} \\
0^{(m-i-1)}\; 1\; 0^{(2i)} & \underset{\wedge}{0}\; 0^{(n-i)}
\end{array}
$$

Here $i = 1$ initially, and the portion beyond $0^{(m-1)}\, 1\, 0^{(2i-2)}\, \underset{\wedge}{0}\, 0^{(n-i+1)}$ takes place only if $i < m$ and $i \leq n$. Thus, if we start with $—\, 1\, 0^{(m)}\, \underset{\wedge}{0}\, 0^{(n)}\, —$, and $n + 1 \geq m$, we end as follows (setting $i = m$):

$$
\begin{array}{r@{\;\;}l}
1\; 1\; 0^{(2m-2)} & \underset{\wedge}{0}\; 0^{(n-m+1)} \\
1\; 1\; 0^{(2m-2)} & \underset{\wedge}{1}\; 0^{(n-m+1)} \\
1 & \underset{\wedge}{1}\; 0^{(2m-2)}\; 1\; 0^{(n-m+1)} \\
1 & \underset{\wedge}{0}\; 0^{(2m-2)}\; 1\; 0^{(n-m+1)} \\
 & \underset{\wedge}{1}\; 0^{(2m-1)}\; 1\; 0^{(n-m+1)} \\
1\; 0^{(2m-1)} & \underset{\wedge}{1}\; 0^{(n-m+1)} \\
1\; 0^{(2m-1)} & \underset{\wedge}{0}\; 0^{(n-m+1)} \\
 & \underset{\wedge}{1}\; 0^{(n+m+1)}
\end{array}
$$

On the other hand, if we start with $— 0^{(m)}\ \underset{\wedge}{0}\ 0^{(n)}\ 1 —$, and $n + 1 < m$ we end as follows (setting $i = n + 1$):

$$0^{(m-n-1)}\ 1\ 0^{(2n)}\ \underset{\wedge}{0}\ 1$$

$$0^{(m-n-1)}\ 1\ 0^{(2n)}\ \underset{\wedge}{1}\ 1$$

$$0^{(m-n-1)}\ \underset{\wedge}{1}\ 0^{(2n)}\ 1\ 1$$

$$0^{(m-n-1)}\ \underset{\wedge}{0}\ 0^{(2n)}\ 1\ 1$$

$$0^{(m-n-2)}\ \underset{\wedge}{0}\ 0^{(2n+1)}\ 1\ 1$$

$$0^{(m-n-2)}\ \underset{\wedge}{1}\ 0^{(2n+1)}\ 1\ 1$$

$$0^{(m-n-2)}\ 1\ 0^{(2n+1)}\ \underset{\wedge}{1}\ 1$$

$$0^{(m-n-2)}\ 1\ 0^{(2n+1)}\ \underset{\wedge}{0}\ 1$$

$$0^{(m-n-2)}\ 1\ 0^{(2n+2)}\ \underset{\wedge}{1}$$

$$0^{(m-n-2)}\ \underset{\wedge}{1}\ 0^{(2n+2)}\ 1$$

$$0^{(m-n-2)}\ \underset{\wedge}{0}\ 0^{(2n+2)}\ 1$$

$$0^{(m+n+1)}\ \underset{\wedge}{1}$$

Definition 1.18. $T_{l\,\mathrm{end}}$ is the following machine:

$$\begin{tikzcd}
\text{Start} \arrow[r] & T_{l\,\mathrm{seek}\,0} \arrow[r] & T_{\mathrm{right}} \arrow[l, bend right, "\text{if } 1"'] \arrow[r, "\text{if } 0"] & T_{\mathrm{left}}
\end{tikzcd}$$

$T_{l\,\mathrm{end}}$ moves the tape to the right until finding 00, and stops on the right-most of these two zeros. $T_{l\,\mathrm{end}}$ does not start counting zeros until moving the tape.

Definition 1.19. $T_{r\,\mathrm{end}}$ is the following machine:

$$\begin{tikzcd}
\text{Start} \arrow[r] & T_{r\,\mathrm{seek}\,0} \arrow[r] & T_{\mathrm{left}} \arrow[l, bend right, "\text{if } 1"'] \arrow[r, "\text{if } 0"] & T_{\mathrm{right}}
\end{tikzcd}$$

$T_{r\,\mathrm{end}}$ moves the tape to the left until finding 00, and stops on the left-most of these two zeros. $T_{r\,\mathrm{end}}$ does not start counting zeros until moving the tape.

Definition 1.20. $T_{l\,\text{trans}}$ is the following machine:

$$\begin{tikzcd}
\text{Start} \arrow[r] & T_{\text{left}} \arrow[r] & T_{\text{left}} \arrow[r, "\text{if } 1"] \arrow[d, "\text{if } 0"] & T_0 \arrow[r] & T_{\text{right}} \arrow[r] & T_1 \arrow[llll, bend right] \\
 & & T_{\text{right}} & & &
\end{tikzcd}$$

The action of $T_{l\,\text{trans}}$ is indicated thus, in the case of interest to us:

$$\begin{array}{c}
\text{————}\ \underset{\wedge}{a}\ 0\ 1^{(b+1)}\ 0\ \text{—} \\
\vdots \\
\text{——}\ a\ 1^{(b+1)}\ \underset{\wedge}{0}\ 0\ \text{————}
\end{array}$$

The tape is otherwise unchanged

Definition 1.21. $T_{l\,\text{shift}}$ is the following machine:

$$\begin{tikzcd}
\text{Start} \arrow[r] & T_{l\,\text{seek}\,0} \arrow[r] & T_{\text{right}} \arrow[r, "\text{if } 1"] \arrow[d, "\text{if } 0"] & T_0 \arrow[r] & T_{l\,\text{trans}} \arrow[lll, bend right] \\
 & & T_{l\,\text{trans}} & &
\end{tikzcd}$$

$T_{l\,\text{shift}}$ acts as follows in the case of interest to us:

$$\begin{array}{c}
\text{——}\ 0\ 1^{(x+1)}\ 0\ 1^{(y+1)}\ \underset{\wedge}{0}\ \text{————} \\
\vdots \\
\text{————}\ 0\ 1^{(y+1)}\ \underset{\wedge}{0}\ 0^{(x+2)}\ \text{——}
\end{array}$$

The tape to the left and right of this portion of $x + y + 5$ symbols is unchanged.

Definition 1.22. T_{fin} is the following machine:

$$\begin{tikzcd}
\text{Start} \arrow[r] & T_{l\,\text{seek}\,0} \arrow[r] & T_{\text{right}} \arrow[r, "\text{if } 1"] \arrow[d, "\text{if } 0"] & T_{\text{left}} \arrow[r] & T_{r\,\text{seek}\,0} \arrow[r] & T_{l\,\text{shift}} \arrow[llll, bend right] \\
 & & T_{l\,\text{trans}} \arrow[d] & & & \\
 & & T_{l\,\text{seek}\,0} \arrow[d] & & & \\
 & & T_{\text{right}} \arrow[d] & & & \\
 & & T_{l\,\text{trans}} & & &
\end{tikzcd}$$

T_{fin} acts as follows in the case of interest to us:

$$\begin{array}{l}
____\ 0^{(2)}\ 1^{(x0+1)}\ 0\ 1^{(x1+1)}\ 0\ \cdots\ 0\ 1^{(x(m-1)+1)}\ 0\ 1^{(y+1)}\ \underset{\wedge}{0}\ ____\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\vdots\\
__________________________\ 1^{(y+1)}\ \underset{\wedge}{0}\ 0^{(p)}\ ____
\end{array}$$

where $p = x_0 + x_1 + \cdots + x_{m-1} + 2m + 2$. In case $m = 0$, it works like this:

$$\begin{array}{l}
____\ 0^{(2)}\ 1^{(y+1)}\ \underset{\wedge}{0}\ ____\\
\qquad\qquad\qquad\qquad\vdots\\
_______\ 1^{(y+1)}\ \underset{\wedge}{0}\ 0^{(2)}\ ____
\end{array}$$

In each case the tape is otherwise unchanged. Here "fin" abbreviates "finish."

This machine will be used at the end of computations to erase scratchwork.

Definition 1.23. T_{copy} is the following machine:

$$\begin{tikzcd}
\text{Start} \arrow[r] & T_{l\,\text{seek}\,0} \arrow[r] & T_{\text{left}} \arrow[r, "\text{if } 1"] \arrow[d, "\text{if } 0"] & T_0 \arrow[r] & T_{r\,\text{seek}\,0} \arrow[r] & T_{r\,\text{seek}\,0} \arrow[r] & T_1 \arrow[r] & T_{l\,\text{seek}\,0} \arrow[r] & T_{l\,\text{seek}\,0} \arrow[r] & T_1 \arrow[lllllll, bend right] \\
& & T_{r\,\text{seek}\,0} & & & & & & &
\end{tikzcd}$$

T_{copy} acts as follows:

$$\begin{array}{l}
_______\ 0\ 1^{(x+1)}\ \underset{\wedge}{0}\ 0^{(x+2)}\ ____\\
\qquad\qquad\qquad\qquad\qquad\vdots\\
___\ 0\ 1^{(x+1)}\ 0\ 1^{(x+1)}\ \underset{\wedge}{0}\ ______
\end{array}$$

The tape is otherwise unchanged.

A machine M repeated $m > 0$ times will be indicated by M^m in our diagrams.

Definition 1.24. For $n > 0$, $T_{n\,\text{copy}}$ is the following machine:

$$\begin{tikzcd}
& & T^{n}_{r\,\text{seek}\,0} & & & & & \\
\text{Start} \arrow[r] & T^{n}_{l\,\text{seek}\,0} \arrow[r] & T_{\text{left}} \arrow[u, "\text{if } 0"] \arrow[r, "\text{if } 1"] & T_0 \arrow[r] & T^{n+1}_{r\,\text{seek}\,0} \arrow[r] & T_1 \arrow[r] & T^{n+1}_{l\,\text{seek}\,0} \arrow[r] & T_1 \arrow[lllll, bend left]
\end{tikzcd}$$

$T_{n\,\mathrm{copy}}$ acts as follows:

$$0\ 1^{(x0+1)}\ 0\ 1^{(x1+1)}\ 0\ \cdots\ 1^{(x(n-1)+1)}\ \underset{\wedge}{0}\ 0^{(x0+2)}$$

$$0\ 1^{(x0+1)}\ 0\ 1^{(x1+1)}\ 0\ \cdots\ 0\ 1^{(x(n-1)+1)}\ 0\ \cdots\ 1^{(x0+1)}\ \underset{\wedge}{0}$$

The tape is otherwise unchanged. This machine copies the nth block to the left.

These are all the basic machines needed to compute functions. We shall return to Turing machines after discussing some classes of number-theoretic functions.

BIBLIOGRAPHY

1. Davis, M. *Computability and Unsolvability*. New York: McGraw-Hill (1958).
2. Hermes, H. *Enumerability, Decidability, Computability*, 2nd ed. New York: Springer (1969).
3. Minsky, M. *Computation*. Englewood Cliffs: Prentice-Hall (1967).

EXERCISES

1.25. Give an example of a Turing machine which gets in a loop—repeats some configurations over and over.

1.26. Give an example of a Turing machine which never stops, but doesn't get in a loop.

1.27′. Prove rigorously that $T_{l\,\mathrm{trans}}$ does what is said in the text.

1.28′. Prove rigorously that $T_{l\,\mathrm{shift}}$ does what is said in the text.

1.29′. Prove rigorously that T_{fin} does what is said in the text.

1.30′. Prove rigorously that T_{copy} does what is said in the text.

1.31′. Prove rigorously that $T_{n\,\mathrm{copy}}$ does what is said in the text.

1.32. Show that there is no Turing machine which, started at an arbitrary position, will find the left-most 1 on the tape.

1.33. Construct a Turing machine which will print the sequence 11001100. . . .

1.34. Construct a Turing machine that stops iff there are at least two one's on the tape.

2 Elementary recursive and primitive recursive functions

To show that many number-theoretic functions are Turing computable, it is convenient to distinguish some functions by closure conditions.

The class of elementary recursive functions which we shall now define in this way is a class of intuitively effective functions which contains most of the effective functions actually encountered in practice. However, not every effective function is elementary recursive. Toward the end of the chapter we introduce the wider class of primitive recursive functions, which still does not cover all kinds of intuitively effective functions. In the next chapter we go from primitive recursive functions to a class of functions, the recursive functions, intuitively corresponding to the entire class of effective functions. An elementary recursive function is just a function obtainable from the usual arithmetic operations of addition, subtraction, multiplication, and division by composition, summation, and multiplication. Most of this chapter is concerned with listing out some elementary functions and with giving operations which lead from elementary functions to elementary functions. This is necessary in order to be able to easily recognize that some of the rather complicated intuitively effective functions are, in fact, elementary recursive. A more detailed treatment of the topics of this section can be found in Péter [2].

Definition 2.1. A *number-theoretic function* is a function which is, for some positive integer m, an m-ary operation on ω. The class of *elementary recursive*, or for brevity *elementary functions*, is the intersection of all classes A of number-theoretic functions such that, first of all, the following specific functions are in A:

(1) $+$, the usual 2-ary operation of addition;
(2) $\cdot$, the usual 2-ary operation of multiplication;

(3) the 2-ary operation f such that $f(m, n) = |m - n|$ for all $m, n \in \omega$;
(4) the 2-ary operation f such that $f(m, n)$ is the greatest nonnegative integer $\leq m/n$ (if $n \neq 0$), 0 if $n = 0$; we denote $f(m, n)$ by $[m/n]$;
(5) for each positive integer n and each $i < n$, the n-ary operation f on ω such that for all $x_0, \ldots, x_{n-1} \in \omega, f(x_0, \ldots, x_{n-1}) = x_i$; f is denoted by U_i^n; it is called an *identity* or *projection* function.

Second, and last, A is required to be closed under the following operations upon number-theoretic functions:

(a) The operation of composition. If f is an m-ary function, and $g_0, \ldots, g_{m-1}$ are n-ary functions, then the *composition* of f with $g_0, \ldots, g_{m-1}$ is denoted by $K_n^m(f; g_0, \ldots, g_{m-1})$; it is defined to be the n-ary function h such that for all $x_0, \ldots, x_{n-1} \in \omega$,

$$h(x_0, \ldots, x_{n-1}) = f(g_0(x_0, \ldots, x_{n-1}), \ldots, g_{m-1}(x_0, \ldots, x_{n-1})).$$

(b) The operation of summation. If f is an m-ary function, then g (m-ary) is obtained from f by *summation*, in symbols $g = \sum f$, if for all $x_0, \ldots, x_{m-1} \in \omega$,

$$g(x_0, \ldots, x_{m-1}) = \sum \{f(x_0, \ldots, x_{m-2}, y) : y < x_{m-1}\}$$

[note that if $m = 1$ the definition reads

$$gx = \sum_{y<x} fy;$$

for any m, we have $g(x_0, \ldots, x_{m-2}, 0) = 0$ by convention].

(c) The operation of multiplication. If f is an m-ary function, then g (m-ary) is obtained from f by *multiplication*, in symbols $g = \prod f$, if for all $x_0, \ldots, x_{m-1} \in \omega$,

$$g(x_0, \ldots, x_{m-1}) = \prod \{f(x_0, \ldots, x_{m-2}, y) : y < x_{m-1}\}$$

[if $x_{m-1} = 0$, the right hand side is 1 by convention].

It should be evident that each elementary function is effectively calculable in the intuitive sense. To convince oneself of this, it is enough to argue that each of the functions (1)–(5) above is effectively calculable, and that the class of effectively calculable functions is closed under the operations (a)–(c). For (1)–(5), the ordinary school algorithms suffice for this argument. As to (a)–(c), suppose, for example, that f, an m-ary function, is effectively calculable, and we wish to show that $\sum f$ also is. Given $x_0, \ldots, x_{m-1} \in \omega$, we merely calculate $f(x_0, \ldots, x_{m-2}, 0), f(x_0, \ldots, x_{m-2}, 1), \ldots, f(x_0, \ldots, x_{m-2}, x_{m-1} - 1)$, which we can do since f is effectively calculable, and then we add them all up by the school process, giving us $(\sum f)(x_0, \ldots, x_{m-1})$.

Proposition 2.2. *Suppose f is m-ary, $g_0, \ldots, g_{m-1}$ are n-ary, and $h_0, \ldots, h_{n-1}$ are p-ary. Then*

$$K_p^n(K_n^m(f; g_0, \ldots, g_{m-1}); h_0, \ldots, h_{n-1}) = K_p^m(f; K_p^n(g_0; h_0, \ldots, h_{n-1}), \ldots, K_p^n(g_{m-1}; h_0, \ldots, h_{n-1})).$$

PROOF. If $x_0, \ldots, x_{p-1} \in \omega$, then, with l = left hand side and r = right hand side,

$$\begin{aligned}
l(x_0, \ldots, x_{p-1}) &= (\mathrm{K}^m_n(f; g_0, \ldots, g_{m-1}))(h_0(x_0, \ldots, x_{p-1}), \ldots, h_{n-1}(x_0, \ldots, x_{p-1})) \\
&= f(g_0(h_0(x_0, \ldots, x_{p-1}), \ldots, h_{n-1}(x_0, \ldots, x_{p-1})), \ldots, \\
&\quad g_{m-1}(h_0(x_0, \ldots, x_{p-1}), \ldots, h_{n-1}(x_0, \ldots, x_{p-1}))) \\
&= f((\mathrm{K}^n_p(g_0; h_0, \ldots, h_{n-1}))(x_0, \ldots, x_{p-1}), \ldots, \\
&\quad (\mathrm{K}^n_p(g_{m-1}; h_0, \ldots, h_{n-1}))(x_0, \ldots, x_{p-1})) \\
&= r(x_0, \ldots, x_{p-1}).
\end{aligned}$$

□

The following theorem is the usual set-theoretical consequence of a definition like 2.1.

Proposition 2.3. *A number-theoretic function f is elementary iff there is a finite sequence $\langle g_0, \ldots, g_{k-1}\rangle$ of number-theoretic functions such that $g_{k-1} = f$, and for each $i < k$ one of the following conditions holds:*

(*i*) $g_i = +$,
(*ii*) $g_i = \cdot$,
(*iii*) g_i = *subtraction* (*in the sense of* 2.1(3)),
(*iv*) g_i = *division* (*in the sense of* 2.1(4)),
(*v*) $g_i = \mathrm{U}^n_j$ *for some* $n > 0$, *some* $j < n$,
(*vi*) g_i *is n-ary, and for some* $m > 0$ *there exist* $j < i$ *and* $k_0, \ldots, k_{m-1} < i$ *such that* g_j *is m-ary*, $g_{k0}, \ldots, g_{k(m-1)}$ *are n-ary, and* $g_i = \mathrm{K}^m_n(g_j; g_{k0}, \ldots, g_{k(m-1)})$ (g_i *is obtained from earlier functions by composition*),
(*vii*) *there is a* $j < i$ *such that* $g_i = \sum(g_j)$,
(*viii*) *there is a* $j < i$ *such that* $g_i = \prod(g_j)$.

PROOF. Let A be the set of all f such that there is a finite sequence of the kind described in the theorem. By considering 1-termed sequences it is easy to see that $+$, $\cdot$, subtraction, division, and U^n_j are all in A (for any $n > 0$ and $j < n$). Suppose $f \in A$, f is m-ary, $h_0, \ldots, h_{m-1} \in A$, all of $h_0, \ldots, h_{m-1}$ are n-ary. Choose a finite sequence $\langle g_0, \ldots, g_{k-1}\rangle$ such that $g_{k-1} = f$ and or each $i < k$ one of the conditions (*i*)–(*viii*) holds for g_i. For each $j < m$ choose a finite sequence $\langle l_{j,0}, \ldots, l_{j,aj-1}\rangle$ such that $l_{j,aj-1} = h_j$ and for each $i < a_j$ one of the conditions (*i*)–(*viii*) holds for l_{ji}. Then consideration of the sequence

$$\langle g_0, \ldots, g_{k-1}, l_{0,0}, \ldots, l_{0,a0-1}, \ldots, l_{m-1,0}, \ldots, \\ l_{m-1,a(m-1)}, \mathrm{K}^m_n(f; h_0, \ldots, h_{m-1})\rangle$$

shows that $\mathrm{K}^m_n(f; h_0, \ldots, h_{m-1}) \in A$. Thus A is closed under composition. If $f \in A$, so that a sequence $\langle g_0, \ldots, g_{k-1}\rangle$ exists as in the theorem, then consideration of $\langle g_0, \ldots, g_{k-1}, \sum f\rangle$ and $\langle g_0, \ldots, g_{k-1}, \prod f\rangle$ show that $\sum f$, $\prod f \in A$. Hence every elementary function appears in A. This proves $\Rightarrow$. If $f \in A$, with $\langle g_0, \ldots, g_{k-1}\rangle$ given as in the theorem, then it is easily shown by induction on i that g_i is elementary for each $i < k$. In particular, $f = g_{k-1}$ is elementary; this proves $\Leftarrow$. ■

We now proceed to show that many garden-variety number-theoretic functions are elementary and that simple operations on elementary functions again give elementary functions.

For later purposes it is convenient to formulate results of the second kind in a more general way. A class A of number-theoretic functions is said to be *closed under elementary recursive operations* provided A contains all the elementary functions 2.1(1)–(5) and is closed under composition, summation, and multiplication. Obviously the class of all elementary functions is closed under elementary recursive operations. So will be all of the wider classes of effective functions which we discuss later.

Proposition 2.4. *Let A be closed under elementary recursive operations. If f is m-ary and $f \in A$, and π is a permutation of $\{0, \ldots, m-1\}$, then the m-ary function g such that $g(x_0, \ldots, x_{m-1}) = f(x_{\pi 0}, \ldots, x_{\pi(m-1)})$ for all $x_0, \ldots, x_{m-1} \in \omega$ is also in A.*

PROOF. $g = \mathrm{K}^m_m(f; \mathrm{U}^m_{\pi 0}, \ldots, \mathrm{U}^m_{\pi m-1)})$. □

Proposition 2.5 (Identification of variables). *Let A be closed under elementary recursive operations. If f is m-ary, $m > 1$, and $f \in A$, then the $(m-1)$-ary function g such that $g(x_0, \ldots, x_{m-2}) = f(x_0, \ldots, x_{m-2}, x_0)$ for all $x_0, \ldots, x_{m-2} \in \omega$ is in A.*

PROOF. $g = \mathrm{K}^m_{m-1}(f; \mathrm{U}^{m-1}_0, \ldots, \mathrm{U}^{m-1}_{m-2}, \mathrm{U}^{m-1}_0)$. □

By means of 2.4 and 2.5 variables can be identified in an arbitrary number of places. Thus if f is 3-ary elementary, so is the function g with $g(x, y) = f(x, y, y)$, for if $h(x, y, z) = f(y, x, z)$, h is elementary by 2.4; letting $k(x, y) = h(x, y, x)$ for all $x, y \in \omega$, k is elementary by 2.5, and $g(x, y) = k(y, x)$ for all $x, y \in \omega$, so g is elementary by 2.4. Usually it is just as easy in cases like this to use the method of proof of 2.4 and 2.5.

Proposition 2.6 (Adjoining apparent variables). *Let A be closed under elementary recursive operations. If f is m-ary and $f \in A$, then the $(m+1)$-ary function g such that $g(x_0, \ldots, x_m) = f(x_0, \ldots, x_{m-1})$ for all $x_0, \ldots, x_m \in \omega$, is in A.*

PROOF. $g = \mathrm{K}^m_{m+1}(f; \mathrm{U}^{m+1}_0, \ldots, \mathrm{U}^{m+1}_{m-1})$. □

Definition 2.7

(*i*) For $n > 0$, $m \in \omega$, C^n_m is the n-ary function such that $\mathrm{C}^n_m(x_0, \ldots, x_{n-1}) = m$ for all $x_0, \ldots, x_{n-1} \in \omega$.

(*ii*) sg and $\overline{\mathrm{sg}}$ are unary functions; for $x \in \omega$,

$$\mathrm{sg}\, x = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x \neq 0, \end{cases}$$

$$\overline{\mathrm{sg}}\, x = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x \neq 0. \end{cases}$$

(*iii*) $\not p$ is a unary function:

$$\not p x = \begin{cases} 0 & \text{if } x = 0 \\ x - 1 & \text{if } x \neq 0 \end{cases} \text{ for all } x \in \omega.$$

(*iv*) By convention, $0^0 = 1$, $0^x = 0$ for $x \neq 0$; $0! = 1$.

(*v*) σ is a unary function: $\sigma x = x + 1$ for all $x \in \omega$.

Thus C^n_m is the n-ary constant function with value m. The functions sg and $\overline{\text{sg}}$ are of a technical usefulness. $\not p$ is the predecessor function and σ the successor function.

Proposition 2.8. *The following functions are elementary:*

(*i*) C^n_m (*for* $n \neq 0$)
(*ii*) σ
(*iii*) sg
(*iv*) $\overline{\text{sg}}$
(*v*) *exponentiation*
(*vi*) *factorial*
(*vii*) $\not p$

PROOF

(1) C^1_0 is elementary: $C^1_0\, x = |x - x|$ for all $x \in \omega$.
(2) $\overline{\text{sg}}$ is elementary: $\overline{\text{sg}}\, x = \prod_{y<x} C^1_0\, y$, for all $x \in \omega$.
(3) sg is elementary: $\text{sg}\, x = \overline{\text{sg}}\, \overline{\text{sg}}\, x$ for all $x \in \omega$.
(4) C^1_1 is elementary: $C^1_1\, x = \overline{\text{sg}}\, C^1_0\, x$ for all $x \in \omega$.
(5) C^1_m is elementary: (by induction on m) $C^1_{m+1}\, x = C^1_m\, x + C^1_1\, x$ for all $x \in \omega$.
(6) C^n_m is elementary: $C^n_m(x_0, \ldots, x_{n-1}) = C^1_m\, U^n_0(x_0, \ldots, x_{n-1})$ for all $x_0, \ldots, x_{n-1} \in \omega$.
(7) $\sigma : \sigma x = x + C^1_1\, x$.
(8) exponentiation: $x^y = \prod_{z<y} U^2_0(x, z)$.
(9) factorial: $x! = \prod_{z<x} \sigma z$.
(10) $\not p : \not p x = |x - C^1_1 x| \cdot \text{sg}\, x$. □

Definition 2.9

(*i*) By an m-ary *number-theoretic relation* ($m > 0$) we mean a set of m-tuples of natural numbers. ${}^m\omega$ is the set of *all* m-tuples of natural numbers. As usual, we identify ${}^1\omega$ and ω, in an informal way.

(*ii*) If R is an m-ary number-theoretic relation, its *characteristic function* χ_R, is the m-ary number-theoretic function such that for all $x_0, \ldots, x_{m-1} \in \omega$,

$$\chi_R(x_0, \ldots, x_{m-1}) = \begin{cases} 0 & \text{if } \langle x_0, \ldots, x_{m-1}\rangle \notin R, \\ 1 & \text{if } \langle x_0, \ldots, x_{m-1}\rangle \in R. \end{cases}$$

(*iii*) An m-ary number-theoretic relation R is *elementary* if χ_R is elementary.

The definition 2.9(*iii*) is motivated by our intuitive feeling that a relation R is effective iff χ_R is an effective function. In fact, if we have an effective procedure for determining membership in R, then we can effectively calculate χ_R as follows. Given any object a, determine whether $a \in R$ or $a \notin R$. In the first case, $\chi_R a = 1$, and in the second case, $\chi_R a = 0$. Conversely, suppose we have an effective procedure for calculating values of χ_R. Given any object a, calculate $\chi_R a$. If $\chi_R a = 1$, then $a \in R$. If $\chi_R a = 0$, then $a \notin R$.

Given any class A of number-theoretic functions, an m-ary number-theoretic relation R is said to be an *A-relation* if $\chi_R \in A$.

Proposition 2.10. *0 and ω are elementary; if $x \in \omega$ then $\{x\}$ is elementary.*

PROOF. $\chi_0 = C_0^1$ and $\chi_\omega = C_1^1$. If $x \in \omega$, then for any $y \in \omega$, $\chi_{\{x\}} y = \overline{\mathrm{sg}}\,(|x - y|)$; hence $\chi_{\{x\}} y = \overline{\mathrm{sg}}\,(|C_x^1\, y - U_0^1\, y|)$. □

By 2.10, $\{x\}$ is always on effectively decidable set. Intuitively speaking, to check whether $y \in \{x\}$ we simply check if $y = x$ (surely an effective matter). As an example, let $B = \{0\}$ if Fermat's last theorem is true, while $B = 0$ if it is false. B is an effectively decidable set, although we do not know now whether $0 \in B$ or not. Thus there is a decision procedure for membership in B, but we don't know what it is (it is either the obvious one for $\{0\}$ or the obvious one for 0).

Proposition 2.11. *Let A be closed under elementary recursive operations. If R and S are A-relations, then so are $R \cap S$, $R \cup S$, and ${}^m\omega \sim R$.*

PROOF. For all $x_0, \ldots, x_{m-1}$, $\chi_{R\cap S}(x_0, \ldots, x_{m-1}) = \chi_R(x_0, \ldots, x_{m-1}) \cdot \chi_S(x_0, \ldots, x_{m-1})$, $\chi_T(x_0, \ldots, x_{m-1}) = \overline{\mathrm{sg}}\, \chi_R(x_0, \ldots, x_{m-1})$, with $T = {}^m\omega \sim R$, $R \cup S = {}^m\omega \sim [({}^m\omega \sim R) \cap ({}^m\omega \sim S)]$. □

Corollary 2.12. *Every finite subset of ω is elementary, and so is every cofinite set.*

Proposition 2.13. *The binary relations $\leq$, $<$, $\geq$, $=$, $\neq$ are elementary.*

PROOF. For any $x, y \in \omega$,

$$\chi_<(x, y) = \overline{\mathrm{sg}}\,[\sigma x / \sigma y] = \overline{\mathrm{sg}}\,[\sigma\, U_0^2\,(x, y) / \sigma\, U_1^2\,(x, y)].$$

Thus $<$ is elementary. Further

$$\chi_{\neq}(x, y) = \mathrm{sg}\,(|x - y|),$$

so $\neq$ is elementary. Finally, $\leq\, = (< \cup =)$, $\geq\, = ({}^2\omega \sim <)$, $>\, = ({}^2\omega \sim \leq)$, $=\, = ({}^2\omega \sim \neq)$. □

Proposition 2.14 (Bounded existential quantifier). *Let A be closed under elementary recursive operations. Suppose R is an m-ary A-relation. Let*

$S = \{\langle x_0, \ldots, x_{m-1}\rangle$: *there is a* $y < x_{m-1}$ *such that* $\langle x_0, \ldots, x_{m-2}, y\rangle \in R\}$. *Then* S *is an* A-*relation.*

PROOF. $\chi_S(x_0, \ldots, x_{m-1}) = \mathrm{sg} \sum \{\chi_R(x_0, \ldots, x_{m-2}, y) : y < x_{m-1}\}$. □

Proposition 2.15 (Bounded universal quantifier). *Let* A *be closed under elementary recursive operations. Suppose* R *is an* m-*ary* A-*relation. Let* $T = \{\langle x_0, \ldots, x_{m-1}\rangle$: *for every* $y < x_{m-1}$ *we have* $\langle x_0, \ldots, x_{m-2}, y\rangle \in R\}$. *Then* T *is an* A-*relation.*

PROOF. Let S be as in 2.14, with R replaced by ${}^m\omega \sim R$. Then $T = {}^m\omega \sim S$. □

Definition 2.16 (Bounded minimum). Let R be an m-ary relation. For all $x_0, \ldots, x_{m-1} \in \omega$, let

$$f(x_0, \ldots, x_{m-1}) = \begin{cases} \text{the least } y < x_{m-1} \text{ such that } \langle x_0, \ldots, x_{m-2}, y\rangle \in R, \\ \quad \text{if there is such a } y, \\ 0 \quad \text{otherwise.} \end{cases}$$

$f(x_0, \ldots, x_{m-1})$ is denoted by $\mu y < x_{m-1} R(x_0, \ldots, x_{m-2}, y)$.

Proposition 2.17. *Let* A *be closed under elementary recursive operations. If* R *is an* m-*ary* A-*relation, then the function* f *of* 2.16 *is a member of* A.

PROOF. Note that

(1) $$\overline{\mathrm{sg}} \sum_{y<i} \chi_R(x_0, \ldots, x_{m-2}, y) = \begin{cases} 1 & \text{if } \langle x_0, \ldots, x_{m-1}, y\rangle \notin R \text{ for all } y < i, \\ 0 & \text{otherwise.} \end{cases}$$

Let $g(x_0, \ldots, x_{m-2}, i) = \overline{\mathrm{sg}} \sum_{y<i} \chi_R(x_0, \ldots, x_{m-2}, y)$ for all $x_0, \ldots, x_{m-2}$, $i \in \omega$. Thus $g \in A$. From (1) we see that

$$\sum \{g(x_0, \ldots, x_{m-2}, \sigma i) ; i < x_{m-1}\} = \begin{cases} f(x_0, \ldots, x_{m-1}) & \text{if there is a } y < \\ \quad x_{m-1} \text{ such that } \langle x_0, \ldots, x_{m-2}, y\rangle \in R, \\ x_{m-1} & \text{otherwise.} \end{cases}$$

Hence

$$f(x_0, \ldots, x_{m-1}) = \overline{\mathrm{sg}}\, g(x_0, \ldots, x_{m-1}) \cdot \sum \{g(x_0, \ldots, x_{m-2}, \sigma i) : i < x_{m-1}\},$$

so $f \in A$. □

The rather technical proof of 2.17 may be compared with a proof of the intuitive version of the proposition, which goes: if R is an m-ary effective relation, then the function f of 2.16 is effective. In fact, to calculate $f(x_0, \ldots, x_{m-1})$, we test successively whether $\langle x_0, \ldots, x_{m-2}, 0\rangle \in R$, $\langle x_0, \ldots, x_{m-2}, 1\rangle \in R, \ldots, \langle x_0, \ldots, x_{m-2}, x_{m-1}\rangle \in R$. If at some point we reach an i such that $\langle x_0, \ldots, x_{m-2}, i\rangle \in R$, we set $f(x_0, \ldots, x_{m-1}) = i$ and stop testing. If we complete our testing without finding such an i we set $f(x_0, \ldots, x_{m-1}) = 0$.

Proposition 2.18 (Definition by cases). *Let* A *be closed under elementary recursive operations. Suppose* $g_0, \ldots, g_{m-1}$ *are* n-*ary members of* A,

$R_0, \ldots, R_{m-1}$ are pairwise disjoint n-ary A-relations with $\bigcup_{i<m} R_i = {}^n\omega$, and f is the n-ary function such that, for all $x_0, \ldots, x_{n-1} \in \omega$,

$$f(x_0, \ldots, x_{n-1}) = \begin{cases} g_0(x_0, \ldots, x_{n-1}) & \text{if } \langle x_0, \ldots, x_{n-1}\rangle \in R_0, \\ g_1(x_0, \ldots, x_{n-1}) & \text{if } \langle x_0, \ldots, x_{n-1}\rangle \in R_1, \\ \cdots\cdots\cdots\cdots & \cdots\cdots\cdots\cdots \\ g_{m-1}(x_0, \ldots, x_{n-1}) & \text{if } \langle x_0, \ldots, x_{n-1}\rangle \in R_{m-1}. \end{cases}$$

Then $f \in A$.

PROOF. For any $x_0, \ldots, x_{n-1} \in \omega$,

$$f(x_0, \ldots, x_{n-1}) = \chi_{R0}(x_0, \ldots, x_{n-1}) \cdot g_0(x_0, \ldots, x_{n-1}) + \cdots + \chi_{R(m-1)}(x_0, \ldots, x_{n-1}) \cdot g_{m-1}(x_0, \ldots, x_{n-1}). \qquad \square$$

Definition 2.19

(*i*) for $x, y \in \omega$, let

$$x \dot{-} y = \begin{cases} x - y & \text{if } x \geq y, \\ 0 & \text{if } x < y. \end{cases}$$

(*ii*)

$$\min(x, y) = \begin{cases} x & \text{if } x \leq y, \\ y & \text{if } x > y. \end{cases}$$

(*iii*) (by induction). For $m > 2$, $\min_m(x_0, \ldots, x_{m-1}) = \min(\min_{m-1}(x_0, \ldots, x_{m-2}), x_{m-1})$, with $\min_2(x, y) = \min(x, y)$.

(*iv*) $\max(x, y)$, $\max_m(x_0, \ldots, x_{m-1})$ similarly.

(*v*) $\mathrm{rm}(x, y)$ = remainder upon dividing x by y, if $y \neq 0$; $\mathrm{rm}(x, 0) = 0$.

(*vi*) $| = \{(x, y) : x \text{ divides } y\} = \{(x, y) : \text{there is a } z \text{ such that } y = x \cdot z\}$.

(*vii*) $\mathrm{PM} = \{x : x \text{ is a positive prime}\}$.

Proposition 2.20. *All of the functions and relations of* 2.19 *are elementary.*

PROOF. Obvious, as concerns (*i*)–(*iv*). For (*v*),

$$\mathrm{rm}(x, y) = \begin{cases} x - (y \cdot [x/y]) & \text{if } y \neq 0, \\ 0 & \text{if } y = 0. \end{cases}$$

For (*vi*), note that $x|y$ iff there is a z such that $y = x \cdot z$ iff there is a $z < y$ such that $y = x \cdot z$; now see 2.14. Finally, $p \in \mathrm{PM}$ iff for every $x < p$, either not $x|p$ or $x = 1$, and $p \neq 0$, $p \neq 1$; cf. 2.15. $\square$

Definition 2.21. For every k let p_k be the $(k + 1)$st prime; thus $\mathrm{p}_0 = 2$, $\mathrm{p}_1 = 3$, $\mathrm{p}_2 = 5, \ldots$.

Proposition 2.22 (Number-theoretic). *For every k, $\mathrm{p}_k \leq \exp(2, 2^k)$.*

PROOF. By induction on k. Trivial for $k = 0, 1$. Induction step, $k > 0$:

$$\begin{aligned} \mathrm{p}_{k+1} &\leq \mathrm{p}_0 \cdot \ldots \cdot \mathrm{p}_k - 1 && \text{(Euclid)} \\ &\leq \exp(2, 2^0) \cdot \ldots \cdot \exp(2, 2^k) - 1 && \text{(induction hypothesis)} \\ &= 2^{\Sigma\{\exp(2,i): i \leq k\}} - 1 \\ &= \exp(2^{k+1} - 1) - 1 \leq \exp(2^{k+1}). \end{aligned} \qquad \square$$

Proposition 2.23. p *is elementary.*

Proof. Let $N = \{(x, y) : x, y \in \mathrm{PM}, x < y$, and y is the next prime after $x\}$. Thus $N = \{(x, y) : x, y \in \mathrm{PM}$ and $x < y$ and for all $z < y$, either $z \leq x$ or not $z \notin \mathrm{PM}\}$, so N is elementary. Let $Pr = \{(x, k) : x$ is the $(k + 1)$st prime$\}$. Thus $(x, k) \in Pr$ if $x \in \mathrm{PM}$ and $\sum_{y<x} \chi_{\mathrm{PM}} y = k$, so Pr is elementary. Finally, $\mathrm{p}_k = \mu x < \exp(2, 2^k) + 1((x, k) \in Pr)$, so p is elementary. □

Definition 2.24. If $a = 0$ or $a = 1$, let $(a)_i = 0$. If $a > 1$ let $(a)_i$ be the exponent of p_i in the prime decomposition of a. Sometimes we write $(a)i$ instead of $(a)_i$.

Proposition 2.25. ( ) *is elementary.*

Proof. $(a)_i = \mu x < a(\mathrm{p}_i^x | a$ and not $\mathrm{p}_i^{x+1} | a)$. □

Definition 2.26. $\mathrm{l}a =$ greatest i such that $\mathrm{p}_i | x$ $(=0$ if $x = 0$ or $1)$.

Proposition 2.27. l *is elementary.*

Proof. $\mathrm{l}a = \mu i < a[\mathrm{p}_i | x$ and $\forall j \leq a(i < j \Rightarrow \mathrm{p}_j \nmid a)]$. □

We now proceed to study a larger class of functions, the class of primitive recursive functions. Most of the effective functions encountered in the literature were actually shown to be primitive recursive. Actually most of them are even elementary, and usually this can easily be shown. We feel that it is only an historical accident that elementary functions are not more widely discussed than primitive recursive functions.

Definition 2.28. The class of *primitive recursive functions* is the intersection of all classes A of functions such that σ, $\mathrm{U}_i^n \in A$ for all $n > 0$ and $i < n$, and such that A is closed under composition and under the following two operations:

(*i*) *The parameterized operation of primitive recursion:* if f is m-ary and h $(m + 2)$-ary, $m > 0$, then define g recursively as follows:

$$g(x_0, \ldots, x_{m-1}, 0) = f(x_0, \ldots, x_{m-1}),$$
$$g(x_0, \ldots, x_{m-1}, \sigma y) = h(x_0, \ldots, x_{m-1}, y, g(x_0, \ldots, x_{m-1}, y)),$$

for all $x_0, \ldots, x_{m-1}, y \in \omega$. Then g *is obtained from f and h by primitive recursion*, in symbols $g = \mathrm{R}^m(f, h)$.

(*ii*) *The no-parameter operation of primitive recursion:* if $a \in \omega$ and h is 2-ary, define g:

$$g0 = a,$$
$$g\sigma y = h(y, gy),$$

for all $y \in \omega$. In symbols $g = \mathrm{R}^0(a, h)$.

A relation R is *primitive recursive* iff χ_R is a primitive recursion function.

Note that the operations of primitive recursion are rather special kinds of recursive or inductive definitions. Many recursive definitions can be reduced to primitive recursive ones; see, e.g., the important course-of-values recursion, 2.33. But there are recursive definitions which cannot be reduced to primitive recursion. See, e.g., Theorem 3.6. The class of general recursive functions introduced in the next section encompasses all of the natural notions of recursive definitions.

Clearly the primitive recursive schema affords an effective procedure for calculating values of $\mathrm{R}^m(f, h)$, if f and h are effectively calculable. Similarly for $\mathrm{R}^0(a, h)$. Thus every primitive recursive function is effectively calculable in the intuitive sense.

Analogously to 2.3 we have:

Proposition 2.29. *A number-theoretic function f is primitive recursive iff there is a finite sequence* $\langle g_0, \ldots, g_{k-1}\rangle$ *of functions such that* $g_{k-1} = f$, *and for each* $i < k$ *one of the following conditions holds:*

(*i*) $g_i = \sigma$,
(*ii*) $g_i = \mathrm{U}_j^n$ *for some* $n > 0, j < n$,
(*iii*) *as in* 2.3 (*vi*) (*composition*),
(*iv*) *there exist* j, $h < i$ *and* $m \in \omega$, $m \neq 0$, *such that* g_j *is m-ary,* g_h *is* $(m + 2)$*-ary, and* $g_i = \mathrm{R}^m(g_j, g_h)$,
(*v*) *there exist* $j < i$ *and* $a \in \omega$ *such that* g_j *is 2-ary and* $g_i = \mathrm{R}^0(a, g_j)$.

A class A of number-theoretic functions is said to be *closed under primitive recursive operations* provided A contains all the primitive recursive functions σ, U_i^n and is closed under the primitive recursive operations given in 2.28, including composition.

Theorem 2.30. *If A is closed under primitive recursive operations, then A is closed under elementary recursive operations. In particular, every elementary function is primitive recursive.*

Proof

(1) $\not{p}$ is primitive recursive. For,

$$\not{p}0 = 0,$$
$$\not{p}\sigma y = \mathrm{U}_0^2(y, \not{p}y).$$

(2) $\dot{-}$ is primitive recursive. For,

$$x \dot{-} 0 = \mathrm{U}_0^1\, x,$$
$$x \dot{-} \sigma y = \not{p}\, \mathrm{U}_2^3\, (x, y, x \dot{-} y).$$

(3) C_0^1 is primitive recursive:

$$\mathrm{C}_0^1\, x = \mathrm{U}_0^1\, x \dot{-} \mathrm{U}_0^1\, x.$$

(4) $+$ is primitive recursive:

$$x + 0 = \mathrm{U}_0^1\, x,$$
$$x + \sigma y = \sigma\, \mathrm{U}_2^3\,(x, y, x + y).$$

(5) $\cdot$ is primitive recursive:

$$x \cdot 0 = \mathrm{C}_0^1\, x,$$
$$x \cdot \sigma y = x \cdot y + x = +(\mathrm{U}_2^3\,(x, y, x \cdot y), \mathrm{U}_0^3\,(x, y, x \cdot y)).$$

(6) $|x - y| = (x \dot{-} y) + (y \dot{-} x)$.

(7) $\overline{\mathrm{sg}}$ is primitive recursive: $\overline{\mathrm{sg}}\, x = 1 \dot{-} x$.

(8) $\mathrm{sg}\,(x) = \overline{\mathrm{sg}}\,\overline{\mathrm{sg}}\, x$.

(9) rm is primitive recursive. Define

$$f(x, 0) = 0,$$
$$f(x, \sigma y) = \sigma f(x, y) \cdot \mathrm{sg}\,|x - \sigma f(x, y)|.$$

Then $\mathrm{rm}\,(x, y) = f(y, x)$.

(10) Division is primitive recursive. Define

$$f(x, 0) = 0,$$
$$f(x, \sigma y) = f(x, y) + \overline{\mathrm{sg}}\,|x - \sigma\, \mathrm{rm}\,(x, y)|.$$

Then $[x/y] = f(y, x)$.

Now assume that A is closed under primitive recursive operations. In particular, A is closed under composition.

(11) A is closed under summation. For, suppose $f \in A$, f m-ary, and $g = \sum f$. Then

$$\begin{aligned} g(x_0, \ldots x_{m-2}, 0) &= 0, \\ g(x_0, \ldots, x_{m-2}, \sigma z) &= \sum_{y < \sigma z} f(x_0, \ldots, x_{m-2}, y) \\ &= \sum_{y < z} f(x_0, \ldots, x_{m-2}, y) + f(x_0, \ldots, x_{m-2}, z) \\ &= g(x_0, \ldots, x_{m-2}, z) + f(x_0, \ldots, x_{m-2}, z). \end{aligned}$$

Hence $g \in A$.

(12) A is closed under multiplication.

This is proved similarly. The proof of 2.30 is complete. □

The converse of 2.30 fails; see 2.45.

To express another important property of primitive recursion, we need a new coding device. Given a finite sequence $\langle x_0, \ldots, x_{m-1}\rangle$ of natural numbers, it is natural to code it by the single integer $\prod_{i<m} \mathrm{p}_i^{x_i+1}$. The added one in the exponents is essential for uncoding, to distinguish between the codes for $\langle 2, 3, 0\rangle$ and $\langle 2, 3\rangle$, for example. The mapping that assigns to each finite sequence of natural numbers its code as above is a one-one function into ω. From the code y the original sequence x is easily extracted:

$$x = \langle (y)_0 \dot{-} 1, \ldots, (y)_{\mathrm{l}y} \dot{-} 1\rangle.$$

The following definition gives a special instance of this coding device:

Definition 2.31. If f is an m-ary number-theoretic function, we define $\tilde{f}$, the *course-of-values function* for f, as follows: $\tilde{f}$ is again m-ary, and for any $x_0, \ldots, x_{m-1} \in \omega$,

$$\tilde{f}(x_0, \ldots, x_{m-1}) = \prod \{\mathrm{p}_i^{f(x0,\ldots,x(m-2),i)+1} : i < x_{m-1}\}.$$

Thus $\tilde{f}(x_0, \ldots, x_{m-1})$ codes the whole sequence $\langle f(x_0, \ldots, x_{m-2}, 0), \ldots, f(x_0, \ldots, x_{m-2}, x_{m-1} - 1)\rangle$. Note that $\tilde{f}(x_0, \ldots, x_{m-2}, 0) = 1$.

Proposition 2.32. *Let A be closed under primitive recursive operations. Then $f \in A$ iff $\tilde{f} \in A$.*

Proof. Assume first that $f \in A$. Then

$$\begin{aligned} \tilde{f}(x_0, \ldots, x_{m-2}, 0) &= 1, \\ \tilde{f}(x_0, \ldots, x_{m-2}, \sigma y) &= \prod_{i < \sigma y} \mathrm{p}_i^{f(x0,\ldots,x(m-2),i)+1} \\ &= \tilde{f}(x_0, \ldots, x_{m-2}, y) \cdot \mathrm{p}_y^{f(x0,\ldots,x(m-2),y)+1} \\ &= h(x_0, \ldots, x_{m-2}, y, \tilde{f}(x_0, \ldots, x_{m-2}, y)), \end{aligned}$$

where $h(z_0, \ldots, z_m) = z_m \cdot \mathrm{p}_{z(m-1)}^{f(z0,\ldots,z(m-1))+1}$ for all $z_0, \ldots, z_m \in \omega$. Conversely, if $\tilde{f} \in A$, then

$$f(x_0, \ldots, x_{m-1}) = (\tilde{f}(x_0, \ldots, x_{m-2}, \sigma x_{m-1}))_{x(m-1)},$$

so $f \in A$. □

The next proposition shows that recursion in which the successor step depends on several preceding values can still be reduced to primitive recursion.

Proposition 2.33 (Course-of-values recursion). *Let A be closed under primitive recursive operations. Suppose f is an m-ary function and h is an $(m + 1)$-ary member of A such that, for all $x_0, \ldots, x_{m-1} \in \omega$,*

$$f(x_0, \ldots, x_{m-1}) = h(x_0, \ldots, x_{m-1}, \tilde{f}(x_0, \ldots, x_{m-1})).$$

Then $f \in A$.

Proof

$$\begin{aligned} \tilde{f}(x_0, \ldots, x_{m-2}, 0) &= 1, \\ \tilde{f}(x_0, \ldots, x_{m-2}, \sigma y) &= \prod_{i < \sigma y} \mathrm{p}_i^{f(x0,\ldots,x(m-2),i)+1} \\ &= \tilde{f}(x_0, \ldots, x_{m-2}, y) \cdot \mathrm{p}_y^{f(x0,\ldots,x(m-2),y)+1} \\ &= \tilde{f}(x_0, \ldots, x_{m-2}, y) \cdot \mathrm{p}_y^{h(x0,\ldots,x(m-2),y,\tilde{f}(x0,\ldots,x(m-2),y))+1} \end{aligned}$$

Thus $\tilde{f} \in A$. By 2.32, $f \in A$. □

Next we show how close elementary functions are to primitive recursive functions—the class of elementary functions is closed under a restricted kind of primitive recursion.

Proposition 2.34 (Bounded primitive recursion)

(*i*) *Suppose* $m > 0$, f *and* h *are elementary, m-ary and* $(m + 2)$*-ary respectively,* $g = \mathrm{R}^m(f, h)$, k *is elementary, and* $g(x_0, \ldots, x_m) \leq k(x_0, \ldots, x_m)$ *for all* $x_0, \ldots, x_m \in \omega$. *Then* g *is elementary.*

(*ii*) *Suppose* h *is a binary elementary function,* $g = \mathrm{R}^0(a, h)$ (*with* $a \in \omega$), *and* $gx \leq kx$ *for all* $x \in \omega$, *with* k *elementary. Then* g *is elementary.*

PROOF. (*i*) For any $x_0, \ldots, x_m, z \in \omega$ let

$$s(x_0, \ldots, x_m) = (x_m + 1) \cdot \sum_{z < xm} k(x_0, \ldots, x_{m-1}, z).$$

Let R consist of all $(m + 2)$-tuples $\langle x_0, \ldots, x_m, y\rangle$ such that there is a $q \leq \mathrm{p}_{xm}^{s(x0,\ldots,xm)}$ so that

$$(1) \qquad (q)_0 = f(x_0, \ldots, x_{m-1})$$

and, for all $z < x_m$,

$$(2) \qquad (q)_{z+1} = h(x_0, \ldots, x_{m-1}, z, (q)_z)$$

and, finally, $y = (q)_{xm}$. Obviously R is elementary. Now (*i*) follows from

$$(3) \qquad g(x_0, \ldots, x_m) = \mu y \leq k(x_0, \ldots, x_m)[\langle x_0, \ldots, x_m, y\rangle \in R].$$

To prove (3), assume that $x_0, \ldots, x_m \in \omega$, let t be the sequence $\langle g(x_0, \ldots, x_{m-1}, 0), \ldots, g(x_0, \ldots, x_{m-1}, x_m)\rangle$, and let

$$q = \prod_{i < xm} \mathrm{p}_i^{ti}.$$

Then for each $i \leq x_m$ we have

$$t_i \leq k(x_0, \ldots, x_{m-1}, i) \leq \sum_{z \leq xm} k(x_0, \ldots, x_{m-1}, z)$$

and so

$$q \leq \mathrm{p}_{xm}^{s(x0,\ldots,xm)}.$$

Furthermore, q satisfies the conditions (1), (2). Thus $\langle x_0, \ldots, x_m, g(x_0, \ldots, x_m)\rangle \in R$. It is also clear that $\langle x_0, \ldots, x_m, y\rangle \in R$ implies that $y = g(x_0, \ldots, x_m)$, so (3) holds.

Condition (*ii*) is proved similarly. □

As our final result of this chapter we shall give an example of a primitive recursive function which is not elementary.

Definition 2.35. α is the binary operation on ω given by the following conditions: for any $m, n \in \omega$,

$$\alpha(m, 0) = m,$$
$$\alpha(m, n + 1) = m^{\alpha(m,n)}.$$

Thus $\alpha(m, n)$ is the iterated exponential, m raised to the m power n times. Although exponentiation is elementary by 2.8(v), we shall see that iterated exponentiation is not. The reason is that it grows faster than any elementary function; see 2.44. Obviously, we have:

Lemma 2.36. *α is primitive recursive.*

Lemma 2.37. *$m \leq \alpha(m, n)$ for all m, n.*

PROOF. We may assume that $m \neq 0$. Now we prove 2.37 by induction on n: $\alpha(m, 0) = m$. Assuming $m \leq \alpha(m, n)$,

$$\alpha(m, n + 1) = m^{\alpha(m,n)} \geq m^m \geq m. \qquad \square$$

Lemma 2.38. *$\alpha(m, n) < \alpha(m, n + 1)$ for all $m > 1$ and all $n \in \omega$.*

PROOF. $\alpha(m, n + 1) = m^{\alpha(m,n)} > \alpha(m, n)$. $\square$

Lemma 2.39. *$\alpha(m, n) < \alpha(m + 1, n)$ for all $m \neq 0$ and all $n \in \omega$.*

PROOF. We proceed by induction on n: $\alpha(m, 0) = m < m + 1 = \alpha(m + 1, 0)$. Assuming our result for n,

$$\begin{aligned} \alpha(m, n + 1) &= m^{\alpha(m,n)} \leq (m + 1)^{\alpha(m,n)} \\ &< (m + 1)^{\alpha(m+1,n)} = \alpha(m + 1, n + 1). \end{aligned} \qquad \square$$

Lemma 2.40. *$\alpha(m, n) + \alpha(m, p) \leq \alpha(m, \max(n, p) + 1)$ for all $m > 1$ and all $n, p \in \omega$.*

PROOF.
$$\begin{aligned} \alpha(m, n) + \alpha(m, p) &\leq 2\alpha(m, \max(n, p)) \text{ by 2.38} \\ &\leq 2^{\alpha(m,\max(n,p))} \leq m^{\alpha(m,\max(n,p))} \\ &= \alpha(m, \max(n, p) + 1). \end{aligned} \qquad \square$$

Lemma 2.41. *$\alpha(m, n) \cdot \alpha(m, p) \leq \alpha(m, \max(n, p) + 1)$ for all $m > 1$ and all $n, p \in \omega$.*

PROOF. If $n = p = 0$ then the inequality is obvious. Hence assume that $n \neq 0$ or $p \neq 0$. Then

$$\begin{aligned} \alpha(m, n) \cdot \alpha(m, p) &\leq \alpha(m, \max(n, p))^2 \quad \text{by 2.38} \\ &= (m^{\alpha(m, \max(n,p)-1)})^2 = m^{2\alpha(m, \max(n,p)-1)} \\ &\leq m^{\exp(2, \alpha(m,\max(n,p)-1))} \leq m^{\alpha(m,\max(n,p))} \\ &= \alpha(m, \max(n, p) + 1). \end{aligned} \qquad \square$$

Lemma 2.42. *$\alpha(m, n)^{\alpha(m,p)} \leq \alpha(m, \max(p + 2, n + 1))$ for all $m > 1$ and all $n, p \in \omega$.*

PROOF. For $n = 0$ we have

$$\alpha(m, n)^{\alpha(m,p)} = m^{\alpha(m,p)} = \alpha(m, p + 1) \leq \alpha(m, \max(p + 2, n + 1))$$

(using 2.38). If $n \neq 0$ we have

$$\begin{aligned} \alpha(m, n)^{\alpha(m,p)} &= m^{\alpha(m,n-1)\cdot\alpha(m,p)} \leq m^{\alpha(m,\max(n-1,p)+1)} \quad \text{by 2.41} \\ &= \alpha(m, \max(p + 2, n + 1)). \end{aligned} \qquad \square$$

Lemma 2.43. *$\alpha(\alpha(m, n), p) \leq \alpha(m, n + 2p)$ for all $m > 1$ and all $n, p \in \omega$.*

PROOF. We proceed by induction on p:

$$\alpha(\alpha(m, n), 0) = \alpha(m, n) = \alpha(m, n + 2\cdot 0).$$

Assuming our result for p, we then have

$$\begin{aligned} \alpha(\alpha(m, n), p + 1) &= \alpha(m, n)^{\alpha(\alpha(m,n),p)} \leq \alpha(m, n)^{\alpha(m,n+2p)} \\ &\leq \alpha(m, \max(n + 2p + 2, n + 1)) \qquad \text{by 2.42} \\ &= \alpha(m, n + 2(p + 1)). \end{aligned}$$

□

Lemma 2.44. *If g is a k-ary elementary function then there is an $m \in \omega$ such that for all $x_0, \ldots, x_{k-1} \in \omega$, if $\max(x_0, \ldots, x_{k-1}) > 1$ then $g(x_0, \ldots, x_{k-1}) < \alpha(\max(x_0, \ldots, x_{k-1}), m)$.*

PROOF. Let A be the set of all functions g (of any rank) for which there is such an m. To prove the lemma it suffices to show that A is closed under elementary recursive operations.

(1) $$+ \in A.$$

In fact, let $m = 2$: for any $x_0, x_1 \in \omega$ with $\max(x_0, x_1) > 1$,

$$\begin{aligned} x_0 + x_1 &\leq \max(x_0, x_1) + \max(x_0, x_1) \\ &= \alpha(\max(x_0, x_1), 0) + \alpha(\max(x_0, x_1), 0) \\ &< \alpha(\max(x_0, x_1), 1) + \alpha(\max(x_0, x_1), 1) \qquad \text{by 2.38} \\ &\leq \alpha(\max(x_0, x_1), 2) \qquad \text{by 2.40} \end{aligned}$$

Thus (1) holds, Analogously,

(2) $$\cdot \in A.$$

(3) $f \in A$, where $f(m, n) = |m - n|$ for all $m, n \in \omega$.

For if $\max(x_0, x_1) > 1$, then $|x_0 - x_1| \leq \max(x_0, x_1) = \alpha(\max(x_0, x_1), 0) < \alpha(\max(x_0, x_1), 1)$. Similarly, the next two statements hold:

(4) $f \in A$, where $f(m, n) = [m/n]$ for all $m, n \in \omega$.

(5) $U^n_i \in A$, for any positive $n \in \omega$ and any $i < n$.

(6) A is closed under composition.

For, suppose f is m-ary, $g_0, \ldots, g_{m-1}$ are n-ary, and $f, g_0, \ldots, g_{m-1} \in A$. Choose $p, q_0, \ldots, q_{m-1} \in \omega$ such that $\max(x_0, \ldots, x_{m-1}) > 1$ implies that $f(x_0, \ldots, x_{m-1}) < \alpha(\max(x_0, \ldots, x_{m-1}), p)$, and such that for each $i < m$, $\max(x_0, \ldots, x_{n-1}) > 1$ implies that $g_i(x_0, \ldots, x_{n-1}) < \alpha(\max(x_0, \ldots, x_{n-1}), q_i)$. Let $h = K^m_n(f; g_0, \ldots, g_{m-1})$. Let

$$s = \max\{q_i : i < m\} + 2p + \max\{f(x_0, \ldots, x_{m-1}) : x_0, \ldots, x_{m-1} \leq 1\} + 1.$$

Now suppose that $\max(x_0, \ldots, x_{n-1}) > 1$. Then if $g_0(x_0, \ldots, x_{n-1}), \ldots, g_{n-1}(x_0, \ldots, x_{n-1}) \leq 1$, we obviously have

$$\begin{aligned} h(x_0, \ldots, x_{n-1}) &= f(g_0(x_0, \ldots, x_{n-1}), \ldots, g_{n-1}(x_0, \ldots, x_{n-1})) \\ &< s \leq \alpha(\max(x_0, \ldots, x_{n-1}), s) \qquad \text{by 2.38} \end{aligned}$$

Assume now that $\max\{g_i(x_0, \ldots, x_{n-1}) : i < m\} > 1$. Then

$$\begin{aligned} h(x_0, \ldots, x_{n-1}) &= f(g_0(x_0, \ldots, x_{n-1}), \ldots, g_{n-1}(x_0, \ldots, x_{n-1})) \\ &< \mathscr{a}(\max\{g_i(x_0, \ldots, x_{n-1}) : i < m\}, p) \\ &< \mathscr{a}(\max\{\mathscr{a}(\max\{x_0, \ldots, x_{n-1}\}, q_i) : i < m\}, p) && \text{by 2.39} \\ &= \mathscr{a}(\mathscr{a}(\max\{x_0, \ldots, x_{n-1}\}, \max\{q_0, \ldots, q_{m-1}\}), p) && \text{by 2.38} \\ &\leq \mathscr{a}(\max\{x_0, \ldots, x_{n-1}\}, \max\{q_0, \ldots, q_{m-1}\} + 2p) && \text{by 2.43} \\ &< \mathscr{a}(\max\{x_0, \ldots, x_{n-1}\}, s) && \text{by 2.38} \end{aligned}$$

(7) A is closed under $\sum$.

In fact, suppose $f \in A$, say f is m-ary, and let $g = \sum f$. Since $f \in A$, choose $p \in \omega$ such that $\max(x_0, \ldots, x_{m-1}) > 1$ implies that $f(x_0, \ldots, x_{m-1}) < \mathscr{a}(\max(x_0, \ldots, x_{m-1}), p)$. Let

$$q = p + 1 + \max\{f(x_0, \ldots, x_{m-1}) : x_0, \ldots, x_{m-1} \leq 1\}.$$

Then for any $x_0, \ldots, x_{m-1} \in \omega, f(x_0, \ldots, x_{m-1}) < \mathscr{a}(\max(x_0, \ldots, x_{m-1}, 2), q)$, using 2.38. Thus if $\max(x_0, \ldots, x_{m-1}) > 1$ we have

$$\begin{aligned} g(x_0, \ldots, x_{m-1}) &= \sum_{y < x(m-1)} f(x_0, \ldots, x_{m-2}, y) \\ &< \sum_{y < x(m-1)} \mathscr{a}(\max(x_0, \ldots, x_{m-2}, y, 2), q) \\ &\leq \sum_{y < x(m-1)} \mathscr{a}(\max(x_0, \ldots, x_{m-1}), q) && \text{by 2.39} \\ &= \mathscr{a}(\max(x_0, \ldots, x_{m-1}), q) \cdot x_{m-1} \\ &\leq \mathscr{a}(\max(x_0, \ldots, x_{m-1}), q) \cdot \mathscr{a}(\max(x_0, \ldots, x_{m-1}), q) && \text{by 2.37} \\ &\leq \mathscr{a}(\max(x_0, \ldots, x_{m-1}), q + 1) && \text{by 2.41} \end{aligned}$$

Similarly, using 2.42,

(8) A is closed under $\prod$.

This completes the proof of 2.44. □

Theorem 2.45. *There are primitive recursive functions which are not elementary in fact,* $\mathscr{a}$ *is such a function.*

Proof. By 2.36, $\mathscr{a}$ is primitive recursive. Suppose $\mathscr{a}$ is elementary. Let $fm = \mathscr{a}(m, m)$ for all $m \in \omega$. Thus f is elementary. By 2.44 choose $m \in \omega$ such that $x > 1$ implies that $fx < \mathscr{a}(x, m)$. Then

$$\begin{aligned} \mathscr{a}(m + 2, m + 2) = f(m + 2) &< \mathscr{a}(m + 2, m) \\ &< \mathscr{a}(m + 2, m + 2) && \text{by 2.38} \end{aligned}$$

contradiction. □

BIBLIOGRAPHY

1. Grzegorczyk, A. Some classes of recursive functions. *Rozprawy Matematyczne, 4* (1953).
2. Péter, R. *Recursive Funktionen.* Berlin: Akademie-Verlag (1957).

EXERCISES

2.46. Show that the following functions are elementary:

(1) $f(x_0, \ldots, x_{m-2}, z) = \begin{cases} \max y \leqslant z((x_0, \ldots, x_{m-2}, y) \in R), \\ =0 \quad \text{if there is no such } y, \end{cases}$

where R is elementary.

(2) $g(x_0, \ldots, x_{m-2}, y) = \max \{f(x_0, \ldots, x_{m-2}, z) : z \leq y\}$, with f elementary.

(3) $g(x_0, \ldots, x_{m-2}, y) = \min \{f(x_0, \ldots, x_{m-2}, z) : z \leq y\}$, with f elementary.

2.47. Show that the following functions and relations are elementary:

(1) $(a, b) =$ gcd (greatest common divisor) of a and b, $=0$ if $a = 0$ or $b = 0$.
(2) $sa =$ sum of positive divisors of a.
(3) the set of perfect numbers, i.e., numbers a with $sa = 2a$.
(4) the Euler φ function: $\varphi a =$ the number of elements of $\{x : 1 \leqslant x \leqslant a\}$ with $(x, a) = 1$.

2.48. Let $fn = [e \cdot n] =$ greatest integer $\leqslant e \cdot n$, for every $n \in \omega$, where e is the base of the natural system of logarithms. Show that f is elementary. *Hint*: write

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!} + \frac{1}{(n+1)!} + \cdots,$$

$$= \frac{1}{n!}\left(n! + \frac{n!}{1!} + \cdots + \frac{n!}{n!}\right) + \frac{1}{(n+1)!} + \cdots.$$

Let $Sn = n! + n!/1! + \cdots + n!/n!$. Define S primitive recursively, but show that is bounded by an elementary function. Let $Rn = 1/(n+1)! + \cdots$ (Note: R is *not* a number-theoretic function, since its values are actually transcendental.) Show that for $n > 1$, $Rn < 1/n!$. Hence conclude that $[e \cdot n] = [Sn/(n-1)!]$ for $n > 1$, as desired.

2.49. Show that $\binom{n}{m}$ (combinatorial symbol) is elementary.

The purpose of the following two exercises is to show how one can be rigorous in applying the results of this section in showing that functions or relations are elementary. However, later we shall not use these exercises, since the application of results of this section are obvious anyway. Both exercises have to do with certain formal languages which are special cases of languages which will be discussed in detail later.

2.50 (Explicit definition). Let A be a class of number-theoretic functions closed under composition, and such that $U_i^n \in A$ whenever $n > 0$ and $i < n$. For each $f \in A$ introduce a symbol R_f. Allow, in addition, variables $v_0, v_1, v_2, \ldots$. We define *term*: any variable standing alone is a term. If $f \in A$, f m-ary ($m > 0$), and $\sigma_0, \ldots, \sigma_{m-1}$ are terms, then so is $R_f\sigma_0, \ldots, \sigma_{m-1}$. These are all the terms.

Let i be such that all the variables appearing in a certain term τ are in the list $v_0, \ldots, v_i$. Define g_τ^i:

$$g_\tau^i(v_0, \ldots, v_i) = \tau$$

for all $v_0, \ldots, v_i \in \omega$, where each R_f occurring in τ is interpreted as f. Show that $g_\tau \in A$. [Try induction on how τ is built up.]

2.51 (COMPLEX EXPLICIT DEFINITION). For each elementary function f introduce a symbol F_f, and for each elementary relation R a symbol $\mathscr{R}_R$. Also let $N_0, N_1, \ldots$ be some more symbols, and $v_0, v_1, \ldots$ variables. For logical symbols we take $\exists, \forall, \mu, \neg, \vee, \wedge, \rightarrow, \leftrightarrow, =$. Special symbols: (,), $<$. We define terms and formulas simultaneously and recursively:

(1) v_i is a term;
(2) if f is an m-ary elementary function and $\sigma_0, \ldots, \sigma_{m-1}$ are terms, then $F_f(\sigma_0, \ldots, \sigma_{m-1})$ is a term;
(3) N_i is a term;
(4) if R is an m-ary elementary relation and $\sigma_0, \ldots, \sigma_{m-1}$ are terms, then $\mathscr{R}_R(\sigma_0, \ldots, \sigma_{m-1})$ is a formula;
(5) if σ, τ are terms then $\sigma = \tau$ is a formula;
(6) if φ and ψ are formulas then so are $\neg\varphi, \varphi \vee \psi, \varphi \wedge \psi, \varphi \rightarrow \psi, \varphi \leftrightarrow \psi$;
(7) if v_i does not occur in a term σ, and if φ is a formula, then $\exists v_i < \sigma, \varphi$ and $\forall v_i < \sigma, \varphi$ are formulas;
(8) under the assumptions of (7), $\mu v_i < \sigma, \varphi$ is a term.

These are all the terms and formulas. Now show:

(9) *if σ is a term whose variables are in the list $v_0, \ldots, v_i$, and if $f_\sigma^i(v_0, \ldots, v_i) = \sigma$ for all $v_0, \ldots, v_i \in \omega$, then f_σ^i is elementary;*
(10) *if φ is a formula whose variables are in the list $v_0, \ldots, v_i$ and if $R_\varphi^i = \{\langle v_0, \ldots, v_i\rangle : v_0, \ldots, v_i \in \omega$ and $\varphi\}$, then R_φ^i is elementary.*

In (9) and (10), the symbol F_f is to be interpreted as f; $\mathscr{R}_R$ as R; N_i as i, and the other symbols are to have their natural meanings.
Suggestion: prove (9), (10) simultaneously by induction on how σ and φ are built up.

2.52. Suppose g and g' are 1-ary primitive recursive and h and h' are 3-ary primitive recursive. Define f and f' simultaneously:

$$f(x, 0) = g(x), \qquad f(x, y + 1) = h(f(x, y), f'(x, y), x)$$
$$f'(x, 0) = g'(x), \qquad f'(x, y + 1) = h'(f(x, y), f'(x, y), x).$$

Show that f and f' are primitive recursive. Hint: define $f''(x, y) = 2^{f(x,y)} \cdot 3^{f'(x,y)}$.

2.53. Suppose that g is 1-ary primitive recursive, h is 4-ary primitive recursive and f is defined as follows:

$$f(0, n) = f(1, n) = gn$$
$$f(m + 1, n) = h(f(m - 1, n), f(m, n), m, n) \qquad \text{for } m > 0.$$

Show that f is primitive recursive.

2.54. Show that there are exactly $\aleph_0$ primitive recursive functions. Show that there is a number-theoretic function which is not primitive recursive.

Recursive Functions; Turing Computability 3

In this chapter we shall give three versions of the notion of effectively calculable function: recursive functions (defined explicitly by means of closure conditions), an analogous but less redundant version due to Julia Robinson, and the notion of Turing computable function, based upon Turing machines. These three notions will be shown to be equivalent; here the results of Chapters 1 and 2 serve as essential lemmas. In the exercises, three further equivalent notions are outlined: a variant of our official definition of recursiveness, the Gödel–Herbrand–Kleene calculus, and a generalized computer version which is even closer to actual computers than Turing machines. As stated in the introduction to this part, none of these different versions stands out as overwhelmingly superior to the others in any reasonable way. The versions involving closure conditions are mathematically the simplest. The ones using generalized machines seem the most intuitively appealing. The Kleene calculus and the Markov algorithms of the next section are closest to the kinds of symbol manipulations and algorithmic procedures that one works out on paper or within natural languages. Take your pick.

Definition 3.1. Let $m > 1$. An m-ary number-theoretic function f is called *special* if for all $x_0, \ldots, x_{m-2} \in \omega$ there is a y such that $f(x_0, \ldots, x_{m-2}, y) = 0$. If f is a special function, we let

$$k(x_0, \ldots, x_{m-2}) = \text{the least } y \text{ such that } f(x_0, \ldots, x_{m-2}, y) = 0.$$

We write "$\mu y(f(x_0, \ldots, x_{m-2}, y) = 0)$" for "$k(x_0, \ldots, x_{m-2})$". The operation of passing from f to k is called the operation of (unbounded) *minimalization*.

The class of *general recursive functions* is the intersection of all classes A of functions such that $\mathscr{s}$, $U_i^n \in A$ for all $n > 0$ and $i < n$, and such that A is closed under composition, primitive recursion, and minimalization (applied to special functions). A relation R is *general recursive* iff χ_R is general recursive. Frequently, both for functions and relations, we shall say merely *recursive* instead of *general recursive*. A class A of number-theoretic functions is said to be *closed under general recursive operations* provided that A contains all the functions σ, U_i^n and is closed under composition, primitive recursion, and minimalization (applied to special functions).

Several comments on Definition 3.1 should be made before we proceed. First, the minimalization operator used in 3.1 is somewhat different from the one in 2.16, and the difference in their notations reflects this. We shall see later that this difference is essential (see, e.g., 3.6). To see that all general recursive functions are effectively calculable it suffices to assume that f is an m-ary special effectively calculable function with $m > 1$ and that k is obtained from f by minimalization and argue that k is effectively calculable. In fact, given $x_0, \ldots, x_{m-2} \in \omega$, start computing $f(x_0, \ldots, x_{m-2}, 0)$, $f(x_0, \ldots, x_{m-2}, 1), \ldots$. Since f is special, 0 eventually appears in this sequence. The first y for which $f(x_0, \ldots, x_{m-2}, y) = 0$ is the desired value of k at $\langle x_0, \ldots, x_{m-2}\rangle$, and the calculation can then terminate. Thus the assumption that f is special is very crucial. Otherwise, for some arguments this procedure would continue forever without yielding an output.

We can argue as follows, intuitively, that every effectively calculable function is general recursive. Let f, m-ary, be effectively calculable. We then have a finitary procedure P to calculate it. Given an argument $\langle x_0, \ldots, x_{m-1}\rangle$, from P we make a calculation c; the last step of the calculation has the value $f(x_0, \ldots, x_{m-1}\rangle$ coded in it. Let T consist of all sequences $\langle P, x_0, \ldots, x_{m-1}, c\rangle$ of this sort. Presumably T itself is effectively calculable and probably more easily calculable than f. By a coding device we may assume that $P \in \omega$ and $c \in \omega$. Let V be the function that finds the output $f(x_0, \ldots, x_{m-1})$ within c. Now it is reasonable to suppose that both T and V are simple enough that they are recursive, for no matter how complicated f is, T and V must be very routinely calculable. Also, it is reasonable to assume that c is uniquely determined by P and $x_0, \ldots, x_{m-1}$. Hence

$$f(x_0, \ldots, x_{m-1}) = V\mu c(\overline{\mathrm{sg}}\,\chi_T(P, x_0, \ldots, x_{m-1}, c) = 0),$$

so f is recursive. We shall see that this intuitive argument is very close to the rigorous argument that every Turing computable function is recursive.

Church's thesis is the philosophical principle that every effectively calculable function is recursive. This principle is important in supplying motivation for our notion of recursiveness. We shall not use it, however, in our formal development. Later, especially in Part III, we shall use what we will call the *weak Church's thesis*, which is just that certain definite arguments and constructions which we shall make are to be seen to be recursive (or even elementary) without a detailed proof. The weak Church's thesis rests on the same foot as the common feeling that most mathematics can be formalized within set theory. Of course we can take extensive practice with checking the weak Church's thesis as strong evidence for Church's thesis itself.

Theorem 3.2. *If A is closed under recursive operations, then A is closed under primitive recursive operations. In particular, every primitive recursive function is recursive.*

Now we want to see that there is a recursive function which is not primitive recursive. The argument which we shall use for this purpose is of some independent interest, so we shall first formulate it somewhat abstractly.

Definition 3.3. Let A be a collection of number-theoretic functions. A binary number-theoretic function f is said to be *universal for unary members of A* provided that for every unary $g \in A$ there is an $m \in \omega$ such that for every $n \in \omega$, $f(m, n) = gn$.

Theorem 3.4. *Let A be a set of number-theoretic functions closed under elementary recursive operations. If f is universal for unary members of A, then $f \notin A$.*

PROOF. Assume that $f \in A$. Let $gm = f(m, m) + 1$ for all $m \in \omega$. Thus $g \in A$. Since f is universal for unary members of A, choose $m \in \omega$ such that $f(m, n) = gn$ for all $n \in \omega$. Then $gm = f(m, m) = f(m, m) + 1$, contradiction. □

The proof just given is an instance of the Cantor diagonal argument. Other instances will play an important role in this part as well as in Part III; see, e.g., 15.18 and 15.20.

Lemma 3.5. *There is a general recursive function which is universal for unary primitive recursive functions.*

PROOF. We first define an auxiliary binary function h by a kind of recursion which is not primitive recursion, and afterwards we will show that h is actually general recursive. We accompany the recursive definition with informal comments. We think of a number x as coding information about an associated primitive recursive function f: $(x)_0$ is the number of arguments of f, and the next prime factor of x indicates in which case of the construction of 2.29 we are in. The definition of $h(x, y)$ for arbitrary $x, y \in \omega$ breaks into the following cases depending upon x:

Case 1 (Successor). $x = 2$. Let $h(x, y) = (y)_0 + 1$ for all y.

Case 2 (Identity functions). $x = 2^n \cdot 3^{i+1}$, where $i < n$. Let $h(x, y) = (y)_i$ for all y.

Case 3 (Composition). $x = 2^n \cdot 5^m \cdot \mathrm{p}_3^q \cdot \mathrm{p}_4^{r0} \cdot \ldots \cdot \mathrm{p}_{m+3}^{r(m-1)}$, with n, $m > 0$. For any y, let

$$h(x, y) = h(q, \mathrm{p}_0^{h(r0,y)} \cdot \ldots \cdot \mathrm{p}_{m-1}^{h(r(m-1),y)}).$$

Note here that $q < x$ and $r0, \ldots, r(m-1) < x$, so the recursion is legal.

Case 4 (Primitive recursion without parameters). $x = 2 \cdot 7^q \cdot 11^a$ with $q > 0$. We define $h(x, y)$ by recursion on y:

$$\begin{aligned} h(x, 1) &= a, \\ h(x, 2^{y+1}) &= h(q, 2^y \cdot 3^{h(x, \exp(2,y))}), \\ h(x, z) &= 0 \qquad \text{for } z \text{ not of the form } 2^u. \end{aligned}$$

Case 5 (Primitive recursion with parameters). $x = 2^{m+1} \cdot 11^q \cdot 13^r$ with $m > 0$ and $q > 0$. We define $h(x, y)$ by recursion on y. First let y be given with $(y)_m = 0$. We set

$$\begin{aligned} h(x, y) &= h(q, y) \\ h(x, y \cdot \mathrm{p}_m^{z+1}) &= h(r, y \cdot \mathrm{p}_m^z \cdot \mathrm{p}_{m+1}^{h(x, y \cdot \exp(pm, z))}). \end{aligned}$$

Case 6. For x not of one of the above forms, let $h(x, y) = 0$ for all y.

This completes the recursive definition of h. We first claim:

(1) for every $m \in \omega \sim 1$ and for every m-ary primitive recursive function f there is an $x \in \omega \sim 1$ such that, for all $y_0, \ldots, y_{m-1} \in \omega$,

$$f(y_0, \ldots, y_{m-1}) = h(x, \mathrm{p}_0^{y0} \cdot \ldots \cdot \mathrm{p}_{m-1}^{y(m-1)}).$$

Indeed, let Γ be the set of all f such that an x exists. Then, for all y,

$$h(2, 2^y) = y + 1,$$

so $\sigma \in \Gamma$. Next, suppose $i < n$. Then for any $y_0, \ldots, y_{n-1} \in \omega$,

$$h(2^n \cdot 3^{i+1}, \mathrm{p}_0^{y0} \cdot \ldots \cdot \mathrm{p}_{n-1}^{y(n-1)}) = y_i,$$

so $\mathrm{U}_i^n \in \Gamma$. To show that Γ is closed under composition, suppose that $f \in \Gamma$, $g_0, \ldots, g_{m-1} \in \Gamma$, f m-ary, and $g_0, \ldots, g_{m-1}$ each n-ary. Choose $u \in \omega$ for f and $v_0, \ldots, v_{m-1} \in \omega$ for $g_0, \ldots, g_{m-1}$ respectively so that (1) holds for f, u; g_0, v_0; $\ldots$; g_{m-1}, v_{m-1}. Let $x = 2^n \cdot 5^m \cdot \mathrm{p}_3^u \cdot \mathrm{p}_4^{v0} \cdot \ldots \cdot \mathrm{p}_{m+3}^{v(m-1)}$. Then for any $y_0, \ldots, y_{n-1} \in \omega$ we have, with $z = \mathrm{p}_0^{y0} \cdot \ldots \cdot \mathrm{p}_{n-1}^{y(n-1)}$, $g_i(y_0, \ldots, y_{n-1}) = t_i$ for each $i < m$,

$$\begin{aligned} h(x, z) &= h(u, \mathrm{p}_0^{h(v0,z)} \cdot \ldots \cdot \mathrm{p}_{m-1}^{h(v(m-1),z)}) \\ &= h(u, \mathrm{p}_0^{t0} \cdot \ldots \cdot \mathrm{p}_{m-1}^{t(m-1)}) \\ &= f(g_0(y_0, \ldots, y_{n-1}), \ldots, g_{m-1}(y_0, \ldots, y_{n-1})). \end{aligned}$$

Thus Γ is closed under composition. To show that Γ is closed under primitive recursion without parameters, suppose $f \in \Gamma$, f binary, with associated number q so that (1) works, and suppose that $x = 2^1 \cdot 7^q \cdot 11^a$. Let $k0 = a$, $k(n + 1) = f(n, kn)$ for all $n \in \omega$. Then we show that $ky = h(x, 2^y)$ for all $y \in \omega$ by induction on y:

$$\begin{aligned} h(x, 2^0) &= a = k0, \\ h(x, 2^{y+1}) &= h(q, 2^y \cdot 3^{h(x, \exp(2,y))}) \\ &= h(q, 2^y \cdot 3^{ky}) \\ &= f(y, ky) = k(y + 1). \end{aligned}$$

It is similarly show that Γ is closed under primitive recursion with parameters. Thus (1) holds.

Now let $f(x, y) = h(x, 2^y)$ for all $x, y \in \omega$. Then by (1), f is universal for unary primitive recursive functions. Hence it only remains to show that h (and hence f) is general recursive. This proof can easily be modified to show that almost any legal kind of recursion leads to a general recursive function. This kind of proof is, however, very laborious. There is a much easier way of proving this kind of thing; see the comments following the recursion theorem in Chapter 5.

The computation of $h(x, y)$ can be done in finitely many steps, in which we compute successively certain other values of h: $h(a_0, b_0), \ldots, h(a_{m-1}, b_{m-1})$. We identify this sequence of computations with the number $\mathrm{p}_0^{c0} \cdot \ldots \cdot \mathrm{p}_{m-1}^{c(m-1)}$, where, for each $i < m$, $ci = 2^{ai} \cdot 3^{bi} \cdot 5^{h(ai,bi)}$. This intuitive idea should be kept in mind in checking the following statement, which clearly shows that h is general recursive. For brevity, we write $(a)_{ij}$ (or $(a)_{i,j}$ or $(a)(i, j)$) in place of $((a)_i)_j$; similar abbreviations hold for $(((a)_i)_j)_k$, etc.

Statement. For any $x, y \in \omega$, $h(x, y) = (z)_{lz,2}$, where z is the least u such that $u \geq 2$, $(u)_{lu,0} = x$, $(u)_{lu,1} = y$, and for each $i \leq lu$ one of the following holds:

(2) $(u)_{i0} = 2$ and $(u)_{i2} = (u)_{i10} + 1$;
(3) $\mathrm{l}(u)_{i0} = 1$ and $(u)_{i01} - 1 < (u)_{i00}$ and $(u)_{i2} = (u)(i, 1, (u)_{i01} - 1)$;
(4) $(u)_{i00} \neq 0$, $(u)_{i01} = 0$, $(u)_{i02} \neq 0$, $\mathrm{l}(u)_{i0} \leq (u)_{i02} + 3$, and there is a $j < i$ such that $(u)_{j0} = (u)_{i03}$, $\mathrm{l}(u)_{j1} \leq (u)_{i02} - 1$, for all $k < (u)_{i02}$ there is a $q < i$ such that $(u)_{q0} = (u)_{i,0,k+4}$, $(u)_{q1} = (u)_{i1}$, and $(u)_{q2} = (u)_{j1k}$, and, finally, $(u)_{j2} = (u)_{i2}$;
(5) $(u)_{i00} = 1$, $(u)_{i01} = (u)_{i02} = 0$, $(u)_{i03} \neq 0$, $\mathrm{l}(u)_{i0} \leq 4$, and one of the following three cases holds:
   (5′) $(u)_{i1} = 1$ and $(u)_{i2} = (u)_{i04}$;
   (5″) there is a $w < (u)_i$ such that $(u)_{i1} = 2^{w+1}$, and there is a $j < i$ such that $(u)_{j0} = (u)_{i03}$ and for some $k < i$, $(u)_{k0} = (u)_{i0}$, $(u)_{k1} = 2^w$, $(u)_{j1} = 2^w \cdot 3^{(u)k2}$, and $(u)_{j2} = (u)_{i2}$;
   (5‴) there is no $w < (u)_i$ such that $(u)_{i1} = 2^w$, and $(u)_{i2} = 0$;
(6) $(u)_{i00} > 1$, $(u)_{i01} = (u)_{i02} = (u)_{i03} = 0$, $(u)_{i04} \neq 0$, $\mathrm{l}(u)_{i0} \leq 5$, and one of the following conditions holds (with $(u)_{i00} - 1 = m$ for brevity):
   (6′) $(u)_{i1m} = 0$ and there is a $j < i$ such that $(u)_{j0} = (u)_{i04}$, $(u)_{j1} = (u)_{i1}$, and $(u)_{j2} = (u)_{i2}$;
   (6″) $(u)_{i1m} \neq 0$, say $(u)_{i1} = t \cdot \mathrm{p}_m$, and there exist $j, k < i$ such that $(u)_{j0} = (u)_{i05}$, $(u)_{j1} = t \cdot \exp(\mathrm{p}_{m+1}, (u)_{k2})$, $(u)_{k0} = (u)_{i0}$, $(u)_{k1} = t$, and $(u)_{j2} = (u)_{i2}$;
(7) none of the above, and $(u)_{i2} = 0$.

To check this statement carefully, let A be the set of all $u \geq 2$ satisfying the condition above beginning "for each $i \leq lu$". Then the following condition is clear:

(8) if $u, v \in A$, then $u \cdot \prod_{i \leq \mathrm{l}v} \mathrm{p}_{\mathrm{l}u+i+1}^{(v)i} \in A$.
(9) for all $x, y \in \omega$, there is a $u \in A$ with $(u)_{\mathrm{l}u,0} = x$ and $(u)_{\mathrm{l}u,1} = y$; for any such u, $(u)_{\mathrm{l}u,2} = h(x, y)$.

Condition (9) is established by induction on x.

This completes the proof of 3.5. □

Theorem 3.6. *There is a recursive function which is not primitive recursive.*

Theorem 3.7. *There are exactly $\aleph_0$ recursive functions.*

Proof. Let A_0 consist of all of the functions $\mathfrak{s}$, U_i^n with $i < n$. Thus $|A_0| = \aleph_0$. Having defined A_n, let A_{n+1} consist of all members of A_n together with all functions obtainable from members of A_n by one application of composition, primitive recursion, or minimalization (applied to special functions). Thus if $|A_n| = \aleph_0$, then $|A_{n+1}| = \aleph_0$. Clearly, then, $|\bigcup_{n\in\omega} A_n| = \aleph_0$. Obviously $\bigcup_{n\in\omega} A_n$ is exactly the set of all recursive functions. □

Theorem 3.8. *There is a number-theoretic function which is not recursive.*

Although Theorem 3.8 follows from 3.7 purely on grounds of cardinality, we can also explicitly exhibit a nonrecursive function. Let $f_0, f_1, \ldots$ be an enumeration of all unary recursive functions (by 3.7). Define $gm = f_m m + 1$ for all $m \in \omega$. Then g is obviously not in our enumeration, so g is not recursive. We are really just repeating the proof of Theorem 3.4 here in a special case.

We now turn to the notion of a Turing computable function.

Definition 3.9

(*i*) If $g = \langle g_0, \ldots, g_{m-1}\rangle$ is a finite sequence of 0's and 1's and F is a tape description (recall Definition 1.2), then we say that *g lies on F beginning at q and ending at n* (where $q, n \in \mathbb{Z}$), provided that $Fq = g0$, $F(q+1) = g_1, \ldots, F_n = g_{m-1}$ (thus $n = q + m - 1$).

(*ii*) An m-ary number-theoretic function f is *Turing computable* iff there is a Turing machine M, with notation as in Definition 1.1, such that for every tape description F, all $q, n \in \mathbb{Z}$, and all $x_0, \ldots, x_{m-1} \in \omega$, if $01^{(x0+1)} 0 \cdots 01^{(x(m-1)+1)}$ lies on F beginning at q and ending at n, and if $Fi = 0$ for all $i > n$, then there is a computation $\langle (F, c_1, n+1), (G_1, a_1, b_1), \ldots, (G_{p-1}, a_{p-1}, b_{p-1})\rangle$ of M having the following properties:

(1) $G_{p-1}i = Fi$ for all $i \leq n + 1$;
(2) $1^{(f(x0,\ldots,x(m-1))+1)}$ lies on G_{p-1} beginning at $n + 2$ and ending at $b_{p-1} - 1$;
(3) $G_{p-1}i = 0$ for all $i \geq b_{p-1}$.

We then say that f is *computed by M*.

There are, of course, several arbitrary aspects in this definition of computable function. Many details could be changed without modifying in an essential way the power of the notion. We have simply specified in a detailed way how an input for the machine is to be presented and how the output is to be located. The condition (1) is particularly useful in combining several computations. Now we show that *every recursive function is Turing computable.*

Lemma 3.10. *σ is Turing computable.*

PROOF. A machine for σ is:

$$T_{\text{copy}} \to T_1 \to T_{\text{left}}.$$ □

Lemma 3.11. *U_i^n is Turing computable.*

PROOF. The machine is $T_{(n-i)\text{copy}}$. □

Lemma 3.12. *The class of Turing computable functions is closed under composition.*

PROOF. Suppose f m-ary, $g_0, \ldots, g_{m-1}$ n-ary. Suppose $f, g_0, \ldots, g_{m-1}$ are computed by $M, N_0, \ldots, N_{m-1}$ respectively. Then the following machine computes $K_n^m(f; g_0, \ldots, g_{m-1})$:

$$\begin{aligned}&T_{\text{left}} \to T_1 \to T_{\text{left}} \to T^n_{(n+1)\text{copy}} \to T^n_{l\,\text{seek}\,0} \to T_{\text{right}} \to T_0 \to T_{\text{rend}} \to\\ &N_0 \to T^n_{(n+1)\text{copy}} \to N_1 \to \cdots \to T^n_{(n+1)\text{copy}} \to N_{m-1} \to T_{(m+(m-1)n)\text{copy}} \to\\ &T_{(m+(m-2)n)\text{copy}} \to \cdots \to T_{m\,\text{copy}} \to M \to T_{\text{fin}}.\end{aligned}$$ ∎

Lemma 3.13. *The class of Turing computable functions is closed under primitive recursion without a parameter.*

PROOF. Suppose that f is a binary operation on ω, computed by a machine M, and $a \in \omega$. Let $g0 = a$, $g(n+1) = f(n, gn)$ for all $n \in \omega$. Then the following machine computes g:

$$T_{\text{left}} \to T_1 \to T_{\text{left}} \to T_{2\,\text{copy}} \to T_{l\,\text{seek}\,0} \to T_{\text{right}} \to T_0 \to T_{\text{rend}} \to$$

$$\begin{tikzcd}
T_{\text{left}} \arrow[r] & (T_1 \to T_{\text{left}})^{a+1} \arrow[r] & T_{2\,\text{copy}} \arrow[r] & T_{\text{right}} \arrow[r] & T_0 \arrow[r] & T_{\text{right}} \arrow[r, "\text{if } 1"] \arrow[d, "\text{if } 0"] & T_{\text{left}} \arrow[r] & {} \\
& & & & & T_{\text{fin}} & &
\end{tikzcd}$$

$$\begin{tikzcd}
T_{\text{left}} \arrow[r] & T_1 \arrow[r] & T_{\text{left}} \arrow[r] & T_{3\,\text{copy}} \arrow[r] & M \arrow[r] & T_{4\,\text{copy}} \arrow[r] & T_{\text{right}} \arrow[r] & T_0 \arrow[r] & T_{\text{right}} \arrow[r, "\text{if } 1"] \arrow[d, "\text{if } 0"] & T_{\text{left}} \arrow[r] & T_{4\,\text{copy}} \arrow[lllllllll, bend right] \\
& & & & & & & & T_{\text{fin}} & &
\end{tikzcd}$$ □

Lemma 3.14. *The class of Turing computable functions is closed under primitive recursion with parameters.*

PROOF. Suppose that f is m-ary, $m > 0$, g is $(m+2)$-ary and that they are computed by M and N respectively. Let $h(x_0, \ldots, x_{m-1}, 0) = f(x_0, \ldots, x_{m-1})$,

$h(x_0, \ldots, x_{m-1}, y + 1) = g(x_0, \ldots, x_{m-1}, y, g(x_0, \ldots, x_{m-1}, y))$. Then the following machine computes h:

$$T_{\text{left}} \to T_1 \to T_{\text{left}} \to T_{2\,\text{copy}} \to T^m_{(m+3)\text{copy}} \to T^{m+1}_{l\,\text{seek}\,0} \to T_{\text{right}} \to$$

$$\begin{tikzcd}[column sep=small]
T_0 \arrow[r] & T_{\text{rend}} \arrow[r] & M \arrow[r] & T_{(m+2)\,\text{copy}} \arrow[r] & T_{\text{right}} \arrow[r] & T_0 \arrow[r] & T_{\text{right}} \arrow[r, "\text{if } 1"] \arrow[d, "\text{if } 0"] & T_{\text{left}} \arrow[r] & {} \\
 & & & & & & T_{\text{fin}} & &
\end{tikzcd}$$

$$\begin{tikzcd}[column sep=small]
T^m_{(m+2)\,\text{copy}} \arrow[r] & T_{\text{left}} \arrow[r] & T_1 \arrow[r] & T_{\text{left}} \arrow[r] & T_{(m+3)\,\text{copy}} \arrow[r] & N \arrow[r] & T_{(m+4)\,\text{copy}} \arrow[r] & T_{\text{right}} \arrow[r, "\text{if } 1"] \arrow[d, "\text{if } 0"] & {} \\
T_{\text{left}} \arrow[r] & T^{m+1}_{(m+4)\,\text{copy}} \arrow[ur] & & & & & & T_{\text{fin}} &
\end{tikzcd}$$

□

Lemma 3.15. *The class of Turing computable functions is closed under minimalization (applied to special functions).*

PROOF. Let f be an m-ary special function, $m > 1$, and suppose that f is computed by a machine M. Let $g(x_0, \ldots, x_{m-2}) = \mu y[f(x_0, \ldots, x_{m-2}, y) = 0]$ for all $x_0, \ldots, x_{m-2} \in \omega$. Then the following machine computes g:

$$\begin{tikzcd}[column sep=small]
T_{\text{left}} \arrow[r] & T_1 \arrow[r] & T_{\text{left}} \arrow[r] & M \arrow[r] & T_{\text{right}} \arrow[r] & T_0 \arrow[r] & T_{\text{right}} \arrow[r, "\text{if } 1"] \arrow[d, "\text{if } 0"] & T_0 \arrow[r] & T_{\text{right}} \arrow[l, bend left, "\text{if } 1"] \arrow[lllllll, bend right, "\text{if } 0"'] \\
 & & & & & & \text{Stop} & &
\end{tikzcd}$$

□

Summarizing Lemmas 3.10–3.15, we have:

Lemma 3.16. *Every general recursive function is Turing computable.*

We now want to get the converse of 3.16. This requires Gödel numbering. This process, whose name is just a catch-word for the process of number-theoretically effectivizing nonnumber-theoretic concepts (already hinted at in the introduction to this part), has already been used twice in less crucial contexts. In discussing course-of-values recursion, we numbered finite sequences of numbers: see 2.31. And in constructing a function universal for unary primitive recursive functions essentially we numbered construction sequences for primitive recursive functions; see 3.5. Now we want to effectively number various of the concepts surrounding the notion of Turing machine. Besides our immediate purpose of proving the equivalence of Turing computability and recursiveness, this effectivization will be important for our later discussion of general recursion theory.

The script letter $\mathscr{g}$ will be used for Gödel numbering functions throughout this book; we will usually just depend on the context to distinguish the various particular uses of "$\mathscr{g}$."

Definition 3.17. Let $\mathbb{E}$ be the set of even numbers. Let T be the class of all Turing machines. *If M is a Turing machine, with notation as in 1.1*, we let the *Gödel number* of M, $\mathcal{g}M$, be the number

$$\prod_{i<2m} \mathrm{p}_i^{ti},$$

where, for each $i < 2m$, $ti = 2^{c[i+2/2]} \cdot 3^{(\chi\mathbb{E})(i+1)} \cdot 5^{v(i+1)} \cdot 7^{d(i+1)}$.

Lemma 3.18. *$\mathcal{g}^*T$ is elementary.*

PROOF. For any $x \in \omega$, $x \in \mathcal{g}^*T$ if $\mathrm{l}x$ is odd, $x > 1$, for every $i \leq \mathrm{l}x$ we have $((x)_i)_2 < 5$, for every $i \leq \mathrm{l}x$ there is a $j \leq \mathrm{l}x$ such that $((x)_i)_3 = ((x)_j)_0$, for every $i \leq \mathrm{l}x$, if i is even then $((x)_i)_0 = ((x)_{i+1})_0$, and for all $i, j \leq \mathrm{l}x$, if $i + 2 \leq j$, then $((x)_i)_0 \neq ((x)_j)_0$, and if i is even then $((x)_i)_1 = 0$, while if i is odd, $((x)_i)_1 = 1$. □

Definition 3.19. If F is a tape description, then the *Gödel number* of F, $\mathcal{g}F$, is the number

$$\prod_{i=0}^{\infty} \mathrm{p}_i^{k_i},$$

where

$$k_i = \begin{cases} F(i/2) & \text{if } i \text{ is even,} \\ F(-(i+1)/2) & \text{if } i \text{ is odd.} \end{cases}$$

Note that a natural number m is the Gödel number of some tape description iff $\forall x < \mathrm{l}m((m)_x < 2)$ and $m \neq 0$.

Definition 3.20. A *complete configuration* is a quadruple (M, F, d, e) such that (F, d, e) is a configuration in the Turing machine M. $\mathbb{C}$ is the set of all complete configurations. The *Gödel number* $\mathcal{g}(M, F, d, e)$ of such a complete configuration is the number

$$2^{\mathcal{g}M} \cdot 3^{\mathcal{g}F} \cdot 5^d \cdot 7^n,$$

where

$$n = \begin{cases} 2e & \text{if } e \geq 0, \\ -2e - 1 & \text{if } e < 0. \end{cases}$$

Lemma 3.21. *$\mathcal{g}^*\mathbb{C}$ is elementary.*

PROOF. For any $x \in \omega$, $x \in \mathcal{g}^*\mathbb{C}$ iff $\forall i \leq \mathrm{l}(x)_1(((x)_1)_i < 2)$, $(x)_1 \neq 0$, $(x)_0 \in \mathcal{g}^*T$, and there is an $i \leq \mathrm{l}(x)_0$ such that $(x)_2 = (((x)_0)_i)_0$, and $\mathrm{l}x \leq 3$. □

Definition 3.22. (i) For any $e \in \mathbb{Z}$, let

$$\mathcal{g}e = \begin{cases} 2e & \text{if } e \geq 0, \\ -2e - 1 & \text{if } e < 0. \end{cases}$$

For any $x \in \omega$, let

$$f_0x = \begin{cases} x+2 & \text{if } x \text{ is even,} \\ 0 & \text{if } x = 1, \\ x-2 & \text{if } x \text{ is odd and } x > 1, \end{cases}$$

$$f_1x = \begin{cases} x-2 & \text{if } x \text{ is even and } x > 0, \\ 1 & \text{if } x = 0, \\ x+2 & \text{if } x \text{ is odd.} \end{cases}$$

Lemma 3.23. *f_0 and f_1 are elementary. For any $e \in \mathbb{Z}$ we have $f_0 \mathscr{g} e = \mathscr{g}(e+1)$ and $f_1 \mathscr{g} e = \mathscr{g}(e-1)$.*

PROOF

$$f_0\mathscr{g}e = \begin{Bmatrix} f_0 2e & e \geq 0 \\ f_0(-2e-1) & e < 0 \end{Bmatrix} = \begin{Bmatrix} 2(e+1) & e \geq 0 \\ 0 & e = -1 \\ -2e-3 & e < -1 \end{Bmatrix} = \mathscr{g}(e+1);$$

$$f_1\mathscr{g}e = \begin{Bmatrix} f_1 2e & e \geq 0 \\ f_1(-2e-1) & e < 0 \end{Bmatrix} = \begin{Bmatrix} 2(e-1) & e > 0 \\ 1 & e = 0 \\ -2e+1 & e < 0 \end{Bmatrix} = \mathscr{g}(e-1). \quad \square$$

Lemma 3.24. *Let $R_0 = \{(x, n, \varepsilon, y) : x = \mathscr{g}F$ for some tape description F, $n = \mathscr{g}e$ for some $e \in \mathbb{Z}$, $\varepsilon = 0$ or $\varepsilon = 1$, and $y = \mathscr{g}(F_\varepsilon^e)\}$. Then R_0 is elementary.*

PROOF. $(x, n, \varepsilon, y) \in R_0$ iff $\forall i \leq \mathrm{l}x((x)_i < 2)$, $x \neq 0$, $\varepsilon < 2$, and $y = [x/\mathrm{p}_n^{(x)n}] \cdot \mathrm{p}_n^\varepsilon$. $\square$

Lemma 3.25. *Let $R_1 = \{(x, y) : x$ is the Gödel number of a complete configuration (M, F, d, e), y is the Gödel number of a complete configuration (M, F', d', e') (same M), and $((F, d, e), (F', d', e'))$ is a computation step$\}$. Then R_1 is elementary.*

PROOF. For any x, y, $(x, y) \in R_1$ iff $x \in \mathscr{g}^*\mathbb{C}$, $y \in \mathscr{g}^*\mathbb{C}$, $(x)_0 = (y)_0$, and there is an $i \leq \mathrm{l}((x)_0)$ such that $(x)_2 = (((x)_0)_i)_0$, $(((x)_0)_i)_1 = ((x)_1)_{(x)3}$, and one of the following conditions holds:

(a) $(((x)_0)_i)_2 = 0$, $((x)_1, (x)_3, 0, (y)_1) \in R_0$, $(y)_2 = (((x)_0)_i)_3$, and $(y)_3 = (x)_3$;
(b) $(((x)_0)_i)_2 = 1$, $((x)_1, (x)_3, 1, (y)_1) \in R_0$, $(y)_2 = (((x)_0)_i)_3$, and $(y)_3 = (x)_3$;
(c) $(((x)_0)_i)_2 = 2$, $(y)_1 = (x)_1$, $(y)_2 = (((x)_0)_i)_3$, and $(y)_3 = f_1((x)_3)$;
(d) $(((x)_0)_i)_2 = 3$, $(y)_1 = (x)_1$, $(y)_2 = (((x)_0)_i)_3$, and $(y)_3 = f_0((x)_3)$. $\square$

Definition 3.26. A *complete computation* is a sequence $\mathfrak{M} = \langle(M, F_0, d_0, e_0), \ldots, (M, F_m, d_m, e_m)\rangle$ such that $\langle(F_0, d_0, e_0), \ldots, (F_m, d_m, e_m)\rangle$ is a computation in M. The *Gödel number* of such a complete computation is the number

$$\prod_{i<m} \mathrm{p}_i^{\mathscr{g}(M, F_i, d_i, e_i)}.$$

Let R_2 be the set of all Gödel numbers of complete computations.

Lemma 3.27. *R_2 is elementary.*

Proof. For any x, $x \in R_2$ iff for every $i \leq \mathrm{l}x$, $(x)_i \in \mathscr{g}^*\mathbb{C}$, and $((((x)_0)_0)_0)_0 = ((x)_0)_2$, and for every $i < \mathrm{l}x$, $((x)_i, (x)_{i+1}) \in R_1$, and there is an $i \leq \mathrm{l}((x)_0)_0$ such that $((((x)_0)_0)_i)_0 = ((x)_{\mathrm{l}x})_2$, $((((x)_0)_0)_i)_1 = (((x)_{\mathrm{l}x})_1)_{((x)_{\mathrm{l}x})_3}$, and

$$((((x)_0)_0)_i)_2 = 4. \qquad \square$$

Definition 3.28. If h is a finite sequence of 0's and 1's, we let

$$\mathscr{g}h = \prod_{i < \mathrm{Dmn}\, h} \mathrm{p}_i^{hi+1}.$$

For any $x \in \omega$, let $f_2 x = \prod_{i \leq x} \mathrm{p}_i^2$.

Lemma 3.29. *f_2 is elementary, and $f_2 x = \mathscr{g}1^{(x+1)}$ for any x.*

Definition 3.30. For any $x, y \in \omega$, $\mathrm{Cat}\,(x, y) = x \cdot \prod_{i \leq \mathrm{l}y} \mathrm{p}_{\mathrm{l}x+i+1}^{(y)_i}$.

Lemma 3.31. *If h and k are finite sequences of* 0's *and* 1's, *then $\mathscr{g}(hk) = \mathrm{Cat}\,(\mathscr{g}h, \mathscr{g}k)$. (Recall the definition of hk from* 1.11.)

Definition 3.32. $f_3^1 x = \mathrm{Cat}\,(2, f_2 x)$. For $m > 1$,

$$f_3^m(x_0, \ldots, x_{m-1}) = \mathrm{Cat}\,(f_3^{m-1}(x_0, \ldots, x_{m-2}), \mathrm{Cat}\,(2, f_2 x_{m-1})).$$

Lemma 3.33. *f_3^m is elementary for each m, and $f_3^m(x_0, \ldots, x_{m-1}) = \mathscr{g}(0\ 1^{(x0+1)}\ 0 \cdots 0\ 1^{(x(m-1)+1)})$.*

Lemma 3.34. *Let $R_3 = \{(x, y, m, n)$: x is the Gödel number of a tape description F, y is the Gödel number of a finite sequence h of* 0's *and* 1's, *$m = \mathscr{g}e$ and $n = \mathscr{g}e'$ for certain $e, e' \in \mathbb{Z}$, and h lies on F beginning at e and ending at $e'\}$. Then R_3 is elementary.*

Proof. For any x, y, m, n $(x, y, m, n) \in R_3$ iff $y \neq 0$, $\forall i \leq \mathrm{l}x((x)_i < 2)$, $x \neq 0$, either $y = 1$ and $m = n$, or else $y > 1$, for every $i \leq \mathrm{l}y[(y)_i = 1$ or $(y)_i = 2]$, and there is a $z \leq (m + 2y)^y$ such that $(z)_0 = m$, $f_0((z)_i) = (z)_{i+1}$ for each $i < \mathrm{l}z$, $\mathrm{l}z = \mathrm{l}y$, $(z)_{\mathrm{l}z} = n$, and for each $i \leq \mathrm{l}z$, $(x)_{(z)i} = (y)_i \dot{-} 1$. $\square$

The notations $f_0, f_1, f_2, f_3^m, R_0, R_1, R_2, R_3$ will not be used beyond the present section. The relations T_m introduced next, however, are fundamental for the aspects of recursion theory dealt with in Chapters 5 and 6.

Definition 3.35. For $m > 0$ let $\mathrm{T}_m = \{(e, x_0, \ldots, x_{m-1}, u)$: e is the Gödel number of a Turing machine M, and u is the Gödel number of a complete computation $\langle (M, F_0, d_0, v_0), \ldots, (M, F_n, d_n, v_n)\rangle$ such that $01^{(x0+1)}\ 0 \cdots 0\ 1^{(x(m-1)+1)}$ lies on F_0 ending at -1, F_0 is zero otherwise, $v_0 = 0$, and $F_n 1 = 1.\}$

Note that for any $e, x_0, \ldots, x_{m-1} \in \omega$ there is at most one u such that $(e, x_0, \ldots, x_{m-1}, u) \in \mathrm{T}_m$.

Lemma 3.36. T_m *is elementary.*

PROOF. For any $e, x_0, \ldots, x_{m-1}, u$, $(e, x_0, \ldots, x_{m-1}, u) \in \mathrm{T}_m$ iff $e \in g^*\mathbb{T}$, $u \in R_2$, $((u)_0)_0 = e$, $(((u)_0)_1, f_3^m(x_0, \ldots, x_{m-1}), s, 1) \in R_3$ and $\forall t \leq ((u)_0)_1$ (t odd and $t > s \Rightarrow (((u)_0)_1)_t = 0$) and $\forall t \leq ((u)_0)_1$ (t even $\Rightarrow (((u)_0)_1)_t = 0$) for some $s \leq ((u)_0)_1$, $((u)_0)_3 = 0$, and $(((u)_{lu})_1)_2 = 1$. □

Definition 3.37. For any $x \in \omega$ let

$$\mathrm{V}x = \mu y \leq x[(((x)_{lx})_1, \mathrm{Cat}\,(f_2 y, 2), 2, 2y + 4) \in R_3]$$

Obviously V is elementary.

Lemma 3.38. *Every Turing computable function is recursive.*

PROOF. Let M be a Turing machine which computes f as described in Definition 3.9(*ii*), and let $e = gM$. Then for any $x_0, \ldots, x_{m-1} \in \omega$,

$$f(x_0, \ldots, x_{m-1}) = \mathrm{V}\mu u[(e, x_0, \ldots, x_{m-1}, u) \in \mathrm{T}_m],$$

as desired. □

Theorem 3.39. *A function is Turing computable iff it is recursive.*

We close this chapter with a variant of the notion of recursiveness due to Julia Robinson [3]. It will be useful to us later on. The idea is to simplify the definition of recursive function by using rather complicated initial functions but very simple recursive operations.

Definition 3.40. $[\sqrt{\ }]$ is the function such that $[\sqrt{x}]$ = greatest integer $\leq \sqrt{x}$ for each $x \in \omega$. Also, for any $x \in \omega$ we let $\mathrm{Exc}\, x = x - [\sqrt{x}]^2$; this is the *excess* of x over a square.

Lemma 3.41. $[\sqrt{\ }]$ *and* Exc *are elementary.*

PROOF. $[\sqrt{x}] \leq x$ for all $x \in \omega$. Further,

$$[\sqrt{0}] = 0$$

$$\begin{aligned}[\sqrt{(n+1)}] &= \begin{cases} [\sqrt{n}] & \text{if } n + 1 \neq ([\sqrt{n}] + 1)^2, \\ [\sqrt{n}] + 1 & \text{otherwise} \end{cases} \\ &= [\sqrt{n}] + \overline{\mathrm{sg}}\, |n + 1 - ([\sqrt{n}] + 1)^2| \end{aligned}$$

Thus we may use 2.34. Finally, $\mathrm{Exc}\, n = n \mathbin{\dot{-}} [\sqrt{n}]^2$. □

The next definition and theorem introduce special cases of the important device of *pairing functions*, extensively used in recursive function theory.

Definition 3.42. (*i*) $J(a, b) = ((a + b)^2 + b)^2 + a$ for all $a, b \in \omega$. (*ii*) $Lx = \text{Exc}\,[\sqrt{x}]$ for all $x \in \omega$.

Theorem 3.43

(*i*) J *and* L *are elementary;*
(*ii*) $\text{Exc}\,0 = 0$ *and* $L0 = 0$;
(*iii*) *if* $\text{Exc}\,(a + 1) \neq 0$, *then* $\text{Exc}\,(a + 1) = \text{Exc}\,a + 1$ *and* $L(a + 1) = La$;
(*iv*) $\text{Exc}\,J(a, b) = a$;
(*v*) $LJ(a, b) = b$;
(*vi*) J *is* $1 - 1$.

Proof. (*i*) and (*ii*) are obvious. As to (*iii*), choose x such that $a = x^2 + \text{Exc}\,a < (x + 1)^2$. Since $\text{Exc}\,(a + 1) \neq 0$, it is then clear that $\text{Exc}\,(a + 1) = \text{Exc}\,a + 1$. Furthermore, clearly $x^2 \leq a < (x + 1)^2$ and $x^2 \leq a + 1 < (x + 1)^2$, so $x = [\sqrt{a}] = [\sqrt{(a + 1)}]$ and hence $La = L(a + 1)$.

To prove (*iv*), note that

$$\begin{aligned} ((a + b)^2 + b)^2 &\leq J(a, b) \\ &< ((a + b)^2 + b)^2 + 2(a + b)^2 + 2b + 1 \\ &= ((a + b)^2 + b + 1)^2. \end{aligned}$$

Hence $\text{Exc}\,J(a, b) = a$, and (*iv*) holds. Furthermore, clearly from the above $[\sqrt{J(a, b)}] = (a + b)^2 + b$; since

$$\begin{aligned} (a + b)^2 &\leq [\sqrt{J(a, b)}] \\ &< (a + b)^2 + 2a + 2b + 1 \\ &= (a + b + 1)^2, \end{aligned}$$

we infer that $LJ(a, b) = \text{Exc}\,[\sqrt{J(a, b)}] = b$, as desired in (*v*). Finally, (*vi*) is a purely set-theoretical consequence of (*iv*) and (*v*). □

For the next results we assume a very modest acquaintance with number theory; see any number theory textbook.

Theorem 3.44 (Number-theoretic: The Chinese remainder theorem). *Let* $m_0, \ldots, m_{r-1}$ *be natural numbers* > 1, *with* $r > 1$, *the* m_i*'s pairwise relatively prime. Let* $a_0, \ldots, a_{r-1}$ *be any* r *natural numbers. Then there is an* $x \in \omega$ *such that* $x \equiv a_i (\text{mod } m_i)$ *for all* $i < r$.

Proof. By induction on r; we first take the case $r = 2$. Since m_0 and m_1 are relatively prime, there exist integers (positive, negative, or zero) s and t such that $1 = m_0 s + m_1 t$. Then $a_0 - a_1 = m_0 s(a_0 - a_1) + m_1 t(a_0 - a_1)$. Choose $u \in \omega$ such that $a_0 - m_0 s(a_0 - a_1) + u m_0 m_1 > 0$, and let $x = a_0 - m_0 s(a_0 - a_1) + u m_0 m_1$. Then $x \equiv a_0 (\text{mod } m_0)$, and $x = a_1 + m_1 t(a_0 - a_1) + u m_0 m_1 \equiv a_1 (\text{mod } m_1)$, as desired.

Now we assume the theorem true for r and prove it with "r" replaced by

"$r + 1$". With s, t, u as above, choose $x \in \omega$ such that $x \equiv a_0 - m_0 s(a_0 - a_1) + u m_0 m_1 (\text{mod } m_0 m_1)$, $x \equiv a_2 (\text{mod } m_2), \ldots, x \equiv a_r (\text{mod } m_r)$. Then $x \equiv a_0 (\text{mod } m_0)$ and, since $a_0 - m_0 s(a_0 - a_1) = a_1 + m_1 t(a_0 - a_1)$, $x \equiv a_1 (\text{mod } m_1)$ as desired. $\square$

Definition 3.45. For all $x, i \in \omega$ let $\beta(x, i) = \text{rm}\,(\text{Exc } x, 1 + (i + 1)\text{L}x)$.

Theorem 3.46 (Number-theoretic: Gödel's β-function lemma). *For any finite sequence $y_0, \ldots, y_{n-1}$ of natural numbers there is an $x \in \omega$ such that $\beta(x, i) = y_i$ for each $i < n$.*

Proof. Let s be the maximum of $y_0, \ldots, y_{n-1}, n$. For each $i < n$ let $m_i = 1 + (i + 1) \cdot s!$ Then for $i < j < n$ the integers m_i and m_j are relatively prime. For, if a prime p divides both m_i and m_j, it also divides $m_j - m_i = (j + 1) \cdot s! - (i + 1) \cdot s! = (j - i) \cdot s!$ Now $p \nmid s!$, since $p | 1 + (i + 1)s!$. Hence $p | j - i$. But $j - i < n \leq s$, and hence this would imply that $p | s!$, which we know is impossible. Thus indeed m_i and m_j are relatively prime.

Hence by the Chinese remainder theorem choose v such that

$$v \equiv y_i (\text{mod } m_i) \qquad \text{for each } i < n.$$

Let $x = \text{J}(v, s!)$. Then $\text{Exc } x = v$ by 3.43(*iv*), and $\text{L}x = s!$ by 3.43(*v*). Hence if $i < n$ we have

$$\begin{aligned} \beta(x, i) &= \text{rm}\,(\text{Exc } x, 1 + (i + 1)\text{L}x) \\ &= \text{rm}\,(v, m_i) \\ &= y_i. \end{aligned}$$

$\square$

Definition 3.47. If f is a 1-place function with range ω, let $f^{(-1)}y = \mu x(fx = y)$ for all $y \in \omega$. We say that $f^{(-1)}$ is obtained from f by *inversion*.

Theorem 3.48 (Julia Robinson). *The class of recursive functions is the intersection of all classes A of functions such that $+$, $\mathfrak{s}$, Exc, $\text{U}_i^n \in A$ (for $0 \leq i < n$), and such that A is closed under the operations of composition, and of inversion (applied to functions with range ω).*

Proof. Clearly the indicated intersection is a subset of the class of recursive functions ($f^{(-1)}y = \mu x(|fx - y| = 0)$, so we have here a special case of minimalization). Now suppose that A is a class with the properties indicated in the statement of the theorem. We want to show that every recursive function is in A. This will take several steps.

The general idea of the proof is this: Inversion is a special case of minimalization, and the general case is obtained from inversion by using pairing functions. Primitive recursion is obtained by representing the computation of a function f as a finite sequence of the successive values of f, coding the sequence into one number using the β function, and selecting that number out by minimalization.

Our proof will begin with some preliminaries, giving a stock of members of A, which leads to the fact that the pairing functions are in A. First note that for any $x \in \omega$, $x^2 \leq x^2 + x < (x + 1)^2$, and hence $\text{Exc}\,(x^2 + x) = x$. Thus

(1) $$\text{Exc has range } \omega.$$

Next,

(2) $$\text{Exc}^{(-1)}(2x) = x^2 + 2x \qquad \text{for all } x \in \omega.$$

For, obviously $\text{Exc}\,(x^2 + 2x) = 2x$. If $\text{Exc}\,(y) = 2x$ with $y < x^2 + 2x$, we may write $y = z^2 + 2x < (z + 1)^2$ and so $z < x$ and hence $(z + 1)^2 = z^2 + 2z + 1 \leq z^2 + 2x < (z + 1)^2$, a contradiction. Thus (2) holds.

Again,

(3) $$\text{Exc}^{(-1)}(2x + 1) = x^2 + 4x + 2 \qquad \text{for all } x \in \omega.$$

For, $(x + 1)^2 = x^2 + 2x + 1 < x^2 + 4x + 2 < x^2 + 4x + 4 = (x + 2)^2$, and hence $\text{Exc}\,(x^2 + 4x + 2) = 2x + 1$. Now suppose $\text{Exc}\,(y) = 2x + 1$, with $y < x^2 + 4x + 2$. Choose z such that $y = z^2 + 2x + 1 < (z + 1)^2$. Then $z^2 + 2x + 1 = y < x^2 + 4x + 2 = (x + 1)^2 + 2x + 1$, and hence $z \leq x$. Hence $(z + 1)^2 = z^2 + 2z + 1 \leq z^2 + 2x + 1 = y < (z + 1)^2$, a contradiction. Thus (3) holds.

From (2) we see that $\text{C}_0^1 x = 0 = \text{Exc} \circ \text{Exc}^{(-1)}(x + x)$ for all $x \in \omega$; hence

(4) $$\text{C}_0^1 \in A.$$

Hence by composition with $\circ$,

(5) $$\text{C}_m^n \in A \qquad \text{for all } n > 0 \text{ and all } m \in \omega.$$

Now let $x \ominus y = \text{Exc}\,(\text{Exc}^{(-1)}(2x + 2y) + 3x + y + 4)$ for all $x, y \in \omega$. Thus

(6) $$\ominus \in A.$$

Now if $x \geq y$, then

$$\begin{aligned}
(x + y + 2)^2 &= (x + y)^2 + 4x + 4y + 4 \\
&\leq (x + y)^2 + 2(x + y) + 3x + y + 4 \\
&= \text{Exc}^{(-1)}(2x + 2y) + 3x + y + 4 \qquad \text{by (2)} \\
&< (x + y)^2 + 6x + 6y + 9 \\
&= (x + y + 3)^2.
\end{aligned}$$

Hence

(7) $$x \ominus y = x - y \qquad \text{if } y \leq x.$$

Let $fx = x^2$ for all $x \in \omega$. Then by (2), (7), $fx = \text{Exc}^{(-1)}(2x) - 2x$ for all $x \in \omega$, so

(8) $$f \in A.$$

Next note that $\text{sg}\, x = \text{Exc} \circ (x^2)$ and $\overline{\text{sg}}\, x = 1 \ominus \text{sg}\, x$ for all $x \in \omega$. Thus

(9) $$\text{sg}, \overline{\text{sg}} \in A.$$

Furthermore,

(10) $$\mathrm{Exc} \circ \mathscr{s} \text{ has range } \omega.$$

For, $\mathrm{Exc}\, \mathscr{s}0 = 0$, and if $x \neq 0$, then $\mathrm{Exc}\, \mathscr{s}(x^2 + x - 1) = x$.

Now using 3.43(*iii*) we see that $\mathscr{p}x = \mathrm{Exc}\,(\mathrm{Exc} \circ \mathscr{s})^{(-1)}(x)$ for all $x \in \omega$. Hence

(11) $$\mathscr{p} \in A.$$

Recall that $\mathbb{E}$ is the set of even numbers. Next we show

(12) $$\chi\mathbb{E}(x) = \mathrm{Exc}\, \mathscr{s}\mathscr{s}\, \mathrm{Exc}^{(-1)}x \qquad \text{for all } x \in \omega.$$

For, if $x = 2y$, then

$$\begin{aligned} \mathrm{Exc}\, \mathscr{s}\mathscr{s}\, \mathrm{Exc}^{(-1)}\, x &= \mathrm{Exc}\, \mathscr{s}\mathscr{s}\, (y^2 + 2y) && \text{by (2)} \\ &= \mathrm{Exc}\, (y^2 + 2y + 2) \\ &= 1; \end{aligned}$$

if $x = 2y + 1$, then

$$\begin{aligned} \mathrm{Exc}\, \mathscr{s}\mathscr{s}\, \mathrm{Exc}^{(-1)}\, x &= \mathrm{Exc}\, \mathscr{s}\mathscr{s}\, (y^2 + 4y + 2) && \text{by (3)} \\ &= \mathrm{Exc}\, (y^2 + 4y + 4) \\ &= 0. \end{aligned}$$

From (12) we have:

(13) $$\chi\mathbb{E} \in A.$$

Now let $gx = 2\, \mathrm{Exc}\, x + \overline{\mathrm{sg}}\, \chi\mathbb{E}x$ for all $x \in \omega$. Thus

(14) $$g \in A.$$

We claim:

(15) $$g \text{ has range } \omega.$$

For, if $x = 2y$ then, since $y^2 + y$ is even, $g(y^2 + y) = 2\, \mathrm{Exc}\, (y^2 + y) = 2y = x$. If $x = 2y + 1$ then, since $(y + 1)^2 + y$ is odd, $g((y + 1)^2 + y) = 2y + 1 = x$.

Let $hx = [x/2] =$ greatest integer $y \leq x/2$ for all $x \in \omega$. Then

(16) $$hx = \mathrm{Exc}\, g^{(-1)}x \qquad \text{for all } x \in \omega, \text{ and hence } h \in A.$$

For, $2\, \mathrm{Exc}\, g^{(-1)}x + \overline{\mathrm{sg}}\, \chi\mathbb{E}g^{(-1)}x = x$ for any x; thus if x is even, then $2\, \mathrm{Exc}\, g^{(-1)}x = x$; while if x is odd, $2\, \mathrm{Exc}\, g^{(-1)}x + 1 = x$, as desired.

For any $x \in \omega$, let $kx = [(\mathrm{Exc}\, \mathscr{p}x)/2] + \mathrm{sg}\, x$. Thus

(17) $$k \in A.$$

Furthermore,

(18) $$k(x^2) = x \qquad \text{for all } x.$$

For, if $x = 0$ the result is obvious. If $x \neq 0$, then $\mathscr{p}x^2 = x^2 - 1 = (x - 1)^2 + 2x - 2$, $\mathrm{Exc}\, \mathscr{p}x^2 = 2x - 2$, and hence $k(x^2) = x$, as desired.

Let $lx = [\sqrt{x}]$ for all $x \in \omega$. Then by (18), $lx = k(x \ominus \text{Exc}\, x)$, so

(19) $$l \in A.$$

Hence by (8) and (19)

(20) $$\text{J}, \text{L} \in A.$$

(21) $$\text{if } x < y, \text{ then } x \ominus y = 3x + y + 3.$$

For,

$$\begin{aligned}(x + y + 1)^2 &= (x + y)^2 + 2x + 2y + 1\\ &< (x + y)^2 + 5x + 3y + 4\\ &< (x + y)^2 + 4x + 4y + 4\\ &= (x + y + 2)^2.\end{aligned}$$

Since $x \ominus y = \text{Exc}\,((x + y)^2 + 2(x + y) + 3x + y + 4)$, (21) now follows.

(22) $\chi_{\geq}(x, y) = \text{sg}\,[(x \ominus y) \ominus (3x + y + 3)]$ for all x, $y \in \omega$, and hence $\chi_{\geq} \in A$.

For, if $x \geq y$ then

$$\begin{aligned}\text{sg}\,[(x \ominus y) \ominus (3x + y + 3)] &= \text{sg}\,[(x - y) \ominus (3x + y + 3)] && \text{by (7)}\\ &= \text{sg}\,(3x - 3y + 3x + y + 3 + 3) && \text{by (21)}\\ &= \text{sg}\,(6x - 2y + 6)\\ &= 1\end{aligned}$$

If $x < y$, then

$$\begin{aligned}\text{sg}\,[(x \ominus y) \ominus (3x + y + 3)] &= (3x + y + 3) \ominus (3x + y + 3) && \text{by (21)}\\ &= 0 && \text{(by (7)}\end{aligned}$$

(23) $$\cdot \in A.$$

For, $x \cdot y = [(((x + y)^2 \ominus x^2) \ominus y^2)/2]$ for all x, $y \in \omega$. Let $m(x, y) = |x - y|$ for all $x, y \in \omega$. Then

(24) $$m \in A,$$

for $m(x, y) = \chi_{\geq}(x, y) \cdot (x \ominus y) + \chi_{\geq}(y, x) \cdot (y \ominus x)$.

With the aid of the auxiliary functions which we have shown to be in A, we can now show how minimalization can be reduced to inversion.

(25) Suppose f is a 2-ary special function, $f \in A$. Let $gx = \mu y(f(x, y) = 0)$, for all $x \in \omega$. Then for all $x \in \omega$, $gx = \text{L}\mu z(f(\text{Exc}\, z, \text{L}z) = 0$, $\text{Exc}\, z + \text{L}z = [\sqrt{[\sqrt{z}]}]$, and $\text{Exc}\, z = x)$ (and for each x there always is a z satisfying the conditions in parentheses).

To prove this, let $x \in \omega$ be given. Let $z = \text{J}(x, gx)$. Then $f(\text{Exc}\, z, \text{L}z) = f(x, gx) = 0$ by 3.43; $\text{Exc}\, z + \text{L}z = x + gx = [\sqrt{[\sqrt{z}]}]$ by direct computation, and $\text{Exc}\, z = x$. Clearly also $\text{L}z = gx$. It remains to show that our choice of z gives the least integer s satisfying the conditions of the μ − operator. Assume that $f(\text{Exc}\, s, \text{L}s) = 0$, $\text{Exc}\, s + \text{L}s = [\sqrt{[\sqrt{s}]}]$, and $\text{Exc}\, s = x$. Say

$s = p^2 + x < (p+1)^2$. Then $[\sqrt{s}] = p$, and $\mathrm{L}s = \mathrm{Exc}\,p$. Say $p = q^2 + \mathrm{L}s < (q+1)^2$. Then $[\sqrt{[\sqrt{s}]}] = q$. Thus $x + \mathrm{L}s = q$. Since $f(\mathrm{Exc}\,s, \mathrm{L}s) = 0$, we have $gx \leq \mathrm{L}s$. Hence $x + gx \leq q$, $(x + gx)^2 \leq q^2$, $(x + gx)^2 + gx \leq p$, $[(x + gx)^2 + gx]^2 \leq p^2$, and $z = \mathrm{J}(x, gx) \leq p^2 + x = s$, as desired. Thus (25) is established.

(26) Under the hypothesis of (25) we have $g \in A$.

For, let $nz = \overline{\mathrm{sg}}\, f(\mathrm{Exc}\, z, \mathrm{L}z) \cdot \overline{\mathrm{sg}}\, (|\mathrm{Exc}\, z + \mathrm{L}z - [\sqrt{[\sqrt{z}]}]|) \cdot \mathrm{Exc}\, z$. By (25), n has range ω, and clearly $n \in A$. Clearly for any $x \in \omega$ we have $gx = \overline{\mathrm{sg}}\, x \cdot g0 + \mathrm{L}n^{(-1)}x$, so $g \in A$.

Next,

(27) if f is special, $f \in A$, and g is obtained from f by minimalization, then $g \in A$.

For suppose f is m-ary, $m > 1$. We proceed by induction on m; the case $m = 2$ is given by (26). Inductively assume that $m > 2$. Define f' by

$$f'(x_0, \ldots, x_{m-2}) = f(\mathrm{Exc}\, x_0, \mathrm{L}x_0, x_1, \ldots, x_{m-2}),$$

for all $x_0, \ldots, x_{m-2} \in \omega$. Clearly f' is special, since f is. Let g' be obtained from f' by minimalization. By the induction hypothesis, $g' \in A$. Now if $x_0, \ldots, x_{m-2} \in \omega$, then

$$\begin{aligned} g(x_0, \ldots, x_{m-2}) &= \mu y(f(x_0, \ldots, x_{m-2}, y) = 0) \\ &= \mu y(f(\mathrm{Exc}\, \mathrm{J}(x_0, x_1), \mathrm{LJ}(x_0, x_1), x_2, \ldots, x_{m-2}, y) = 0) \\ &= \mu y(f'(\mathrm{J}(x_0, x_1), x_2, \ldots, x_{m-2}, y) = 0) \\ &= g'(\mathrm{J}(x_0, x_1), x_2, \ldots, x_{m-2}); \end{aligned}$$

hence $g \in A$ by (20).

For all $x, y \in \omega$ let $g(x, y) = [x/y]$.

(28) $g \in A$.

For, if $x, y \in \omega$ then

$$\begin{aligned} [x/y] &= \mu z(y \cdot \sigma z > z \text{ or } y = 0) \\ &= \mu z(\chi_{\geq}(x, y \cdot \sigma z) \cdot y = 0), \end{aligned}$$

so $g \in A$ by (27) and (22).

Now since $\mathrm{rm}\,(x, y) = x \ominus ([x/y] \cdot y)$, we have

(29) $\mathrm{rm} \in A$.

Hence by (20),

(30) $\beta \in A$.

Now we can take care of primitive recursion.

Suppose g is obtained from f and h by primitive recursion, f m-ary and h $(m+2)$-ary, $m > 0$. Then for any $x_0, \ldots, x_{m-1}, y \in \omega$,

(31) $g(x_0, \ldots, x_{m-1}, y) = \beta(\mu z[\beta(z, 0) = f(x_0, \ldots, x_{m-1})$ and $\mu w(\beta(z, \sigma w) \neq h(x_0, \ldots, x_{m-1}, w, \beta(z, w))$ or $w = y) = y], y)$;

such z and w always exist, for any $x_0, \ldots, x_{m-1}, y \in \omega$.

To prove (31), let $x_0, \ldots, x_{m-1}, y \in \omega$ be given. By Theorem 3.46 choose z such that $\beta(z, i) = g(x_0, \ldots, x_{m-1}, i)$ for each $i \leq y$. Thus if $\sigma w \leq y$ we have

$$\begin{aligned} \beta(z, \sigma w) &= g(x_0, \ldots, x_{m-1}, \sigma w) \\ &= h(x_0, \ldots, x_{m-1}, w, g(x_0, \ldots, x_{m-1}, w)) \\ &= h(x_0, \ldots, x_{m-1}, w, \beta(z, w)). \end{aligned}$$

Hence

$$\mu w(\beta(z, \sigma w) \neq h(x_0, \ldots, x_{m-1}, w, \beta(z, w)) \text{ or } w = y) = y.$$

Furthermore, $\beta(z, 0) = g(x_0, \ldots, x_{m-1}, 0) = f(x_0, \ldots, x_{m-1})$. Hence there is a z of the sort mentioned in (31). Let t be the least such z. By induction on i it is easily seen that for any $i \leq y$ we have $\beta(t, i) = g(x_0, \ldots, x_{m-1}, i)$. Hence $\beta(t, y) = g(x_0, \ldots, x_{m-1}, y)$, as desired.

(32) Under the hypothesis of (31), if in addition f and h are in A, then $g \in A$.

For, first let

$$k'(x_0, \ldots, x_{m-1}, y, z, w) = \overline{\text{sg}}\,|\beta(z, \sigma w) - h(x_0, \ldots, x_{m-1}, w, \beta(z, w))| \cdot \text{sg}(|w - y|)$$

for all $x_0, \ldots, x_{m-1}, y, z, w \in \omega$. Then $k' \in A$ by (9), (24), and (30). Furthermore, obviously k' is special and

$$\begin{aligned} g(x_0, \ldots, x_{m-1}, y) = \beta(\mu z[\beta(z, 0) = f(x_0, \ldots, x_{m-1}) \quad &\text{and} \\ \mu w(k'(x_0, \ldots, x_{m-1}, y, z, w) = 0) &= y], y) \end{aligned}$$

Let $k''(x_0, \ldots, x_{m-1}, y, z) = \mu w(k'(x_0, \ldots, x_{m-1}, y, z, w) = 0)$ for all $x_0, \ldots, x_{m-1}, y, z \in \omega$. Then $k'' \in A$ by (27). Let $k'''(x_0, \ldots, x_{m-1}, y, z) = \text{sg}(|\beta(z, 0) - f(x_0, \ldots, x_{m-1})|) + \text{sg}(|k''(x_0, \ldots, x_{m-1}, y, z) - y|)$. Then $k''' \in A$, and by (31) k''' is special; moreover,

$$g(x_0, \ldots, x_{m-1}, y) = \beta(\mu z(k'''(x_0, \ldots, x_{m-1}, y, z) = 0), y).$$

Hence $g \in A$, as desired.

(33) If g is obtained from a and h by primitive recursion, $a \in \omega$ and h binary and $h \in A$, then $g \in A$.

The proof is similar to that of (32).

Thus σ, $U_i^n \in A$, and A is closed under composition, primitive recursion, and minimalization (applied to special functions). Hence, every recursive function is in A, and the proof of 3.48 is complete. □

Definition 3.49. Let P be the two-place operation on one place functions such that

$$P(f, g)(x) = fx + gx$$

for all one place functions f, g and all $x \in \omega$.

Theorem 3.50. *The class of* 1*-place recursive functions is the intersection of all sets* A *of* 1*-place functions such that* $\mathfrak{s}$, $\mathrm{Exc} \in A$ *and* A *is closed under* K_1^1, P, *and inversion (applied to functions with range* ω*).*

PROOF. Clearly the intersection indicated is included in the class of 1-place recursive functions. Now suppose that A satisfies the conditions of the theorem. Note that $\mathrm{U}_0^1 \in A$, since $\mathrm{U}_0^1 = \mathrm{K}_1^1(\mathrm{Exc}, \mathrm{Exc}^{(-1)})$. If f is a 1-place recursive function, then $f = \mathrm{K}_1^1(f, \mathrm{U}_0^1)$. Hence in order to show that all 1-place recursive functions are in A (which is all that remains for the proof), it suffices to prove the statement

(*) if f is an m-ary general recursive function and $g_0, \ldots, g_{m-1} \in A$, then $\mathrm{K}_1^m(f; g_0, \ldots, g_{m-1}) \in A$.

To prove (*), let B be the set of all f such that if f is m-ary and $g_0, \ldots, g_{m-1} \in A$ then $\mathrm{K}_1^m(f; g_0, \ldots, g_{m-1}) \in A$. Note that for f unary we have $f \in A$ iff $f \in B$. Hence $+$, $\mathfrak{s}$, Exc, $\mathrm{U}_i^n \in B$ (for $0 \le i < n < \omega$) and B is closed under inversion, applied to functions with range ω. To show that B is closed under composition, assume that f (m-ary) is in B, that $h_0, \ldots, h_{m-1}$ (all n-ary) are in B, and that $g_0, \ldots, g_{n-1} \in A$. Then by 2.2,

$$\begin{aligned}\mathrm{K}_1^n(\mathrm{K}_n^m(f; h_0, \ldots, h_{m-1}); g_0, \ldots, g_{n-1}) \\ = \mathrm{K}_1^m(f; \mathrm{K}_1^n(h_0; g_0, \ldots, g_{n-1}), \ldots, \mathrm{K}_1^n(h_{m-1}; g_0, \ldots, g_{n-1})).\end{aligned}$$

Now $\mathrm{K}_1^n(h_0; g_0, \ldots, g_{n-1}), \ldots, \mathrm{K}_1^n(h_{m-1}; g_0, \ldots, g_{n-1}) \in A$, so

$$\mathrm{K}_1^n(\mathrm{K}_n^m(f; h_0, \ldots, h_{m-1}); g_0, \ldots, g_{n-1}) \in A.$$

Thus, $g_0, \ldots, g_{n-1}$ being arbitrary, $\mathrm{K}_n^m(f; h_0, \ldots, h_{m-1}) \in B$. Hence by 3.48 the proof is complete. □

BIBLIOGRAPHY

1. Davis, M. *Computability and Unsolvability*. New York: McGraw-Hill (1958).
2. Hermes, H. *Enumerability, Decidability, Computability*. New York: Springer (1969).
3. Robinson, J. General recursive functions. *Proc. Amer. Math. Soc.*, *1* (1950), 703–718.

EXERCISES

3.51. Show that the set Γ in the proof of 3.5 is closed under primitive recursion with parameters.

3.52*. Let $f(0, y) = y + 1$, $f(x + 1, 0) = f(x, 1)$, and $f(x + 1, y + 1) = f(x, f(x + 1, y))$. Show that f is recursive.

3.53*. (continuing **3.52***). Show that f is not primitive recursive. *Hint:* prove the following in succession (for any $x, y \in \omega$):

(1) $y < f(x, y)$;
(2) $f(x, y) < f(x, y + 1)$;

(3) $f(x, y + 1) \leq f(x + 1, y)$;
(4) $f(x, y) < f(x + 1, y)$;
(5) $f(1, y) = y + 2$;
(6) $f(2, y) = 2y + 3$;
(7) for any $c_1, \ldots, c_r$ there is a d such that for all x, $\Sigma_{1 \leq j \leq r} f(c_j, x) \leq f(d, x)$ (prove first for $r = 2$, taking $d = \max(c_1, c_2) + 4$);
(8) for every primitive recursive function g (say with n places) there is a c such that for all $x_1, \ldots, x_n$, $g(x_1, \ldots, x_n) < f(c, x_1 + \cdots + x_n)$;
(9) f is not primitive recursive.

3.54. What difficulty would arise in deleting "primitive" from Lemma 3.5 [show that 3.5 would then be false, but also indicate how a proof roughly similar to that given for 3.5 would break down].

3.55. Express a Turing machine to compute $+$ directly in terms of the machines of Chapter 1, i.e., don't use results of this section.

3.56. The set $\{gM : M$ is a Turing machine with exactly five states$\}$ is elementary.

3.57′. Prove (33) in the proof of 3.48 in detail.

3.58. The class of recursive functions is the intersection of all classes A of functions such that σ, $U_i^n \in A$ (each $n > 0$, each i with $i < n$), $+ \in A$, $\dot{-} \in A$, $\cdot \in A$, and A is closed under composition and under minimalization (applied to special functions). Thus we have here another equivalent definition of the notion of recursive function; this version, or slight variations of it, are frequently found in the literature.

3.59. Let $J_1(x, y) = 2^x.(2y + 1) - 1$ for all $x, y \in \omega$, let $K_1x = (x + 1)_0$, and let $L_1x = ([(x + 1)/\exp(2, K_1x)] - 1)/2$. Then show:

(1) J_1, K_1, L_1 are elementary
(2) $J_1(K_1x, L_1x) = x$
(3) $K_1J_1(x, y) = x$
(4) $L_1J_1(x, y) = y$
(5) J_1 maps $\omega \times \omega$ $1-1$ onto ω

3.60. For any $x, y \in \omega$, let $J_2(x, y) = [(x + y)^2 + 3x + y]/2$, $Q_1x = [([\sqrt{(8x + 1)}] + 1)/2] \dot{-} 1$, $Q_2x = 2x - (Q_1x)^2$, $K_2x = (Q_2x - Q_1x)/2$, and $L_2x = Q_1x - K_2x$. Prove analogs of 3.59(1)–(5). *Hint*: Define $f: \omega \times \omega \rightarrow \omega \times \omega$ by putting

$$f(x, y) = \begin{cases} (x + 1, y - 1) & \text{if } y \neq 0, \\ (0, x + 1) & \text{if } y = 0. \end{cases}$$

(The function f describes a certain easily visualized procedure of going through all pairs (x, y).)

Prove that $J_2f(x, y) = J_2(x, y) + 1$ for all $x, y \in \omega$. Thus J_2 is the natural mapping $\omega \times \omega \rightarrow \omega$ associated with f. Then show successively that J_2 maps $\omega \times \omega$ onto ω and that for all $x, y \in \omega$, $Q_1J_2(x, y) = x + y$, $Q_2J_2(x, y) = 3x + y$, $K_2J_2(x, y) = x$, $L_2J_2(x, y) = y$. Then $J_2(K_2x, L_2x) = x$ follows easily since J_2 is onto.

3.61*. If f is a 1-place number-theoretic function, we define f^n (temporary notation) by induction:

$$f^0x = x \qquad \text{for all } x \in \omega,$$
$$f^{n+1}x = ff^nx \qquad \text{for all } x \in \omega.$$

The function g such that $gn = f^n0$ for all $n \in \omega$ is said to be obtained from f by *iteration.*

Prove the following theorem:

Theorem (R. M. Robinson). *The class of primitive recursive functions is the intersection of all classes A of functions such that $\mathfrak{s}$, Exc, $+$, $U_i^n \in A$ whenever $i < n \in \omega$ and A is closed under composition and iteration.*

Hint: As in the proof of 3.48 the essential thing is to show that each primitive recursive function is in A, where A satisfies the conditions of the theorem. Proceed stepwise:

(1) C_0^n, sg, $\overline{\text{sg}} \in A$.
(2) Let $fx = x + 2\,\overline{\text{sg}}\,\text{Exc}\,(x + 4) + 1$. Then $f \in A$.
(3) Let $gx = x + 2[\sqrt{x}]$. Then $g \in A$.
(4) Let $hx = x^2$. Then $h \in A$.
(5) Let $x \ominus y = \text{Exc}\,[(x + y)^2 + 3x + y + 1]$. Then $\ominus \in A$.
(6) Let $\alpha x = \overline{\text{sg}}\, x + 2\,\overline{\text{sg}}\,(x \ominus 1)$. Then $\alpha \in A$.
(7) Let β be obtained from α by iteration, $\gamma x = x + 1 + \beta x$, ε obtained from γ by iteration, $kx = [x/2]$ for all x; then $3 > \beta x \equiv x \pmod 3$, and $k_x = \varepsilon x - x$, and $k \in A$.
(8) Let $ix = [\sqrt{x}]$. Then $i \in A$.
(9) $\cdot$, J, $L \in A$.
(10) Suppose $j \in A$, and k is defined from j as follows:

$$k0 = 0,$$
$$k(n + 1) = j(n, kn).$$

Then $k \in A$. *Hint:* define $k'n = J(n, kn)$ for all $n \in \omega$.
(11) Suppose $f_1 \in A$, and f_2 is defined from f_1 as follows:

$$f_2(a, 0) = a,$$
$$f_2(a, n + 1) = f_1(n, f_2(a, n)).$$

Then $f_2 \in A$. *Hint:* define $l0 = 0$, $l(n + 1) = f_2(\text{L}n, \text{Exc}n)$.
(12) Suppose $f_1, f_2 \in A$, and f_3 is defined as follows:

$$f_3(a, 0) = f_1a,$$
$$f_3(a, n + 1) = f_2(n, f_3(a, n)).$$

Then $f_3 \in A$. *Hint:* define $l(a, 0) = a$, $l(a, n + 1) = J(a, f_3(a, n))$
(13) If $f_4, f_5 \in A$, and f_6 is defined by:

$$f_6(a, 0) = f_4a,$$
$$f_6(a, n + 1) = f_5(a, n, f_6(a, n)),$$

then $f_6 \in A$.
(14) A is closed under primitive recursion.

3.62. Using 3.61, show that the class of all 1-ary primitive recursive functions is the intersection of all classes A such that σ, Exc $\in A$ and A is closed under iteration, K_1^1, and P.

3.63* (HERBRAND–GÖDEL–KLEENE CALCULUS). We outline another equivalent version of recursiveness. We need a small formal system:
Variables: $v_0, v_1, v_2, \ldots$;
Individual constant: $\mathbf{0}$;
Operation symbols: $\mathbf{f}_m$ (m-ary), $\mathbf{g}_{mn}$ (m-ary) for all $m \in \omega \sim 1$, $n \in \omega$; σ (unary).

By induction we define $\triangle 0 = \mathbf{0}$, $\triangle(m + 1) = \sigma \triangle m$ for all $m \in \omega$; we denote $\triangle m$ sometimes by $\mathbf{m}$. Now we define terms:

(1) $\langle v_i \rangle$;
(2) $\langle \mathbf{0} \rangle$;
(3) if σ is a term, so is $\sigma\sigma$;
(4) if $m \in \omega \sim 1$ and $\sigma_0, \ldots, \sigma_{m-1}$ are terms, so are $\mathbf{f}_m\sigma_0 \cdots \sigma_{m-1}$ and $\mathbf{g}_{mn}\sigma_0 \cdots \sigma_{m-1}$ for each $n \in \omega$;
(5) terms are formed only in these ways.

An *equation* is an expression $\sigma \equiv \tau$ with σ, τ terms.
A *system of equations* is a finite sequence of equations. If E is a system of equations, say $E = \langle \varphi_0, \ldots, \varphi_{m-1} \rangle$, then an *E-derivation* is a finite sequence $\langle \psi_0, \ldots, \psi_{n-1} \rangle$ of equations such that for each $i < n$ one of the following holds:

(6) $\exists j < m\, \psi_i = \varphi_j$;
(7) $\exists j < i \exists$ variable $\alpha \exists\, m \in \omega$ (ψ_i is obtained from ψ_j by replacing each occurrence of α in ψ_j by $\mathbf{m}$);
(8) $\exists j, k < i\, \psi_k$ has the form $\sigma \equiv \tau$, ψ_j has the form $\mathbf{f}_p\mathbf{x}_0 \cdots \mathbf{x}_{p-1} = \mathbf{y}$ or $\mathbf{g}_{pq}\mathbf{y}_0 \cdots \mathbf{x}_{p-1} = \mathbf{y}$, and ψ_k is obtained from ψ_k by replacing one occurrence of $\mathbf{f}_p\mathbf{x}_0 \cdots \mathbf{x}_{p-1}$ (or $\mathbf{g}_{pq}\mathbf{x}_0 \cdots \mathbf{x}_{p-1}$) in τ by $\mathbf{x}$.

We write $E \vdash \chi$ to mean that there is an E-derivation with last member χ.

Now an m-place number-theoretic function k is called *Herbrand–Gödel–Kleene* recursive if there is a system E of equations such that $\forall x_0 \cdots \forall x_{m-1} \forall y (E \vdash f_m\mathbf{x}_0 \cdots \mathbf{x}_{m-1} = \mathbf{y}$ iff $k(x_0, \ldots, x_{m-1}) = y)$.

Show that k is Herbrand–Gödel–Kleene recursive iff it is recursive.

Hint: To show that every recursive function is HGK recursive, let A be the collection of all functions k (say k is m-ary) such that there is a set E of equations and an assignment of n-ary operations to the n-ary operation symbols occurring in members of E (for all $n \in \omega$), k assigned to $\mathbf{f}_m$, under which all members of E become intuitively true for any values assigned to the variables and such that $\forall x_0 \cdots \forall x_{m-1} \exists y$ ($E \vdash \mathbf{f}_m\mathbf{x}_0 \cdots \mathbf{x}_{m-1}$ $\mathbf{y}$). Show that A satisfies the conditions of Exercise 3.58 and hence that every recursive function is Herbrand–Gödel–Kleene recursive.

To show the converse, do a Gödel numbering. Let $T'_m = \{(e, x_0, \ldots, x_{m-1}, u)$: e is the Gödel number of a system E of equations and u is the Gödel number of an E-derivation with last term of the form $\mathbf{f}_m\mathbf{x}_0 \cdots \mathbf{x}_{m-1} = \mathbf{y}\}$. Given such a u, $\mathrm{V}'u$ is the y mentioned. Then see the proof of 3.38.

3.64* (INFINITE DIGITAL COMPUTER). Yet another equivalent form of recursiveness is obtained by generalizing a first-generation digital computer. We visualize our computer as an infinite array of storage boxes, labeled 0, 1, 2, ... Each storage box is allowed to hold any natural number. By convention we assume that all but finitely many of the boxes have 0 in them. Box 0 is the *instruction counter*. Box 1 is the *accumulator*. All other boxes are just fast memory cells. We supply only six instructions:

(1) add one to the contents of Box 1;
(2) subtract one from the contents of Box 1, or leave zero if already 0;
(3) replace the contents of storage n by the contents of storage 1 (for any n);
(4) replace the contents of storage 1 by the contents of storage n (for any n);
(5) (for each $n \in \omega$) if storage 1 has a zero in it, take the next instruction from storage n otherwise proceed as usual;
(6) stop.

For technical reasons there is no *start* instruction.

The machine works as follows. We set the storages initially to certain values (programming). Then the machine starts. It looks at box 0 and takes its instruction from the box specified there (each instruction will be assigned a number). After performing the instruction, the instruction counter advances one step (except possibly for instructions (5) and (6)), and then the next instruction is executed, etc. The machine continues until hitting the stop instruction, and then stops. It is possible that the machine will get in a "loop", and never stop.

An initial state of the machine is called a *program*. A program computes a 1-place function f as follows. We put x in storage 2 and press the start button. The machine grinds away, and finally stops; fx is then in the accumulator.

Now we express all of this rigorously. A storage description or *program* is a function F mapping ω into ω such that for some $m \in \omega$ we have $Fn = 0$ for all $n \geq m$.

An *instruction* is a number of the form $2^0 3^0$, $2^1 \cdot 3^0$, $2^2 \cdot 3^n$, $2^3 \cdot 3^n$, $2^4 \cdot 3^n$, 0, where $n \in \omega$. These instructions correspond to (1)–(6) above, respectively.

A *computation step* is a pair (F, G) such that F and G are storage descriptions and one of the following conditions holds:

(1) $FF0 = 2^0 \cdot 3^0$ and $G = (F^0_{F0+1})^1_{F1+1}$
(2) $FF0 = 2^1 \cdot 3^0$ and $G = (F^0_{F0+1})^1_{F1 \dot{-} 1}$
(3) $FF0 = 2^2 \cdot 3^n$ (for some n), and $G = (F^0_{F0+1})^n_{F1}$
(4) $FF0 = 2^3 \cdot 3^n$ (for some n), and $G = (F_{F0+1})^1_{Fn}$
(5) $FF0 = 2^4 \cdot 3^n$ (for some n), and

$$G = F^0_{F0+1} \qquad \text{if } F1 \neq 0,$$
$$G = F^0_n \qquad \text{if } F1 = 0.$$

A computation is a finite sequence $\langle F_0, \ldots, F_{m-1} \rangle$ of storage descriptions, with $m > 0$, such that (F_i, F_{i+1}) is a computation step for each $i < m - 1$ and $F_{m-1}F_{m-1}0 = 0$. We say that $\langle F_0, \ldots, F_{m-1} \rangle$ is a computation *beginning with* F_0 and *ending with* F_{m-1}. Now an m-ary function f is said to be *infinite-digital computed* by a program F provided that for all $x_0, \ldots, x_{m-1}$ there is a computation beginning with $F^{2 \ldots m+1}_{x0 \ldots x(m-1)}$ and ending with a program G such that $G1 = f(x_0, \ldots, x_{m-1})$.

Show that a function is infinite-digital computable iff it is recursive.

Markov Algorithms 4*

The present chapter is optional; it is devoted to another important and widely used version of effectiveness, Markov algorithms.

The theory of Markov algorithms is described carefully and in detail in Markov [3]. Here we shall only give enough of its development to prove equivalence with Turing computability and recursiveness. The equivalence was first proved in Detlovs [2]. For a brief outline of the theory see Curry [1].

Definition 4.1. Throughout this chapter, by a *word* we shall understand a finite sequence of 0's, 1's, and 2's. The empty word is admitted. A *Markov algorithm* is a matrix A of the form

$$\begin{matrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ \vdots & \vdots & \vdots \\ a_m & b_m & c_m \end{matrix}$$

such that $a_0, \ldots, a_m, b_0, \ldots, b_m$ are words and $c_0, \ldots, c_m \in \{0, 1\}$. A word a *occurs* in a word b if there are words c and d such that $b = cad$. Of course a may occur in b several times. An *occurrence* of a *in* b is a triple (c, a, d) such that $b = cad$. It is called the *first occurrence of a in b* if c has shortest length among all occurrences of a in b.

An *algorithmic step under A* is a pair (d, e) of words with the following properties:

(*i*) there is an $i \leq m$ such that a_i occurs in d;
(*ii*) if $i \leq m$ is minimum such that a_i occurs in d, and if (f, a_i, g) is the first occurrence of a_i in d, then $e = fb_ig$.

Such an algorithmic step is said to be *nonterminating*, if with i as in (*ii*), $c_i = 0$; otherwise (i.e., with $c_i = 1$), it is called *terminating*. A *computation* *under* A is a finite sequence $\langle d_0, \ldots, d_m\rangle$ of words such that for each $i < m - 1$, (d_i, d_{i+1}) is a nonterminating algorithmic step, while (d_{m-1}, d_m) is a terminating algorithmic step.

Now an m-ary function f is *algorithmic* if there is a Markov algorithm A as above such that for any $x_0, \ldots, x_{m-1} \in \omega$ there is a computation $\langle d_0, \ldots, d_n\rangle$ under A such that the following conditions hold:

(*iii*) $d_0 = 0\ 1^{(x0+1)}\ 0\ \cdots\ 0\ 1^{(x(m-1)+1)}\ 0\ 2$;

(*iv*) $\langle 2\rangle$ occurs only once in d_n;

(*v*) $0\ 1^{(f(x0,\ldots,x(m-1))+1)}\ 0\ 2$ occurs in d_n.

We then say that *A computes f*.

A row $a_i\ b_i\ 0$ in a Markov algorithm will be indicated $a_i \to b_i$, while a row $a_i\ b_i\ 1$ will be indicated $a_i \to \cdot b_i$. A Markov algorithm lists out finitely many substitutions of one word for another, and an algorithmic computation consists in just mechanically applying these substitutions until reaching a substitution of the form $a_i \to \cdot b_i$. Clearly, then, an algorithmic function is effective in the intuitive sense. Markov algorithms are related to Post systems and to formal grammars. Now we shall give some examples of algorithms, which we shall not numerate since they are not needed later. The algorithm A_0:

$$\langle 0\rangle \to \cdot\langle 0\rangle$$

works as follows: any computation under A is of length 2 and simply repeats the word: $\langle a, a\rangle$, where $\langle 0\rangle$ occurs in a. Consider the algorithm A_1:

$$\langle 0\rangle \to \cdot\langle 01\rangle.$$

Some examples of computations under A_1 are:

(1) $\langle\langle 0\rangle, \langle 01\rangle\rangle$

(2) $\langle\langle 00\rangle, \langle 010\rangle\rangle$

(3) $\langle\langle 11010\rangle, \langle 110110\rangle\rangle$.

Let A_2 be the following algorithm:

$$\begin{aligned}\langle 0\rangle &\to \langle 1\rangle\\ \langle 1\rangle &\to \cdot\langle 1\rangle.\end{aligned}$$

The algorithm A_2 takes any word and replaces all 0's by 1's, then stops. Let A_3 be

$$\langle 1\rangle \to \langle 11\rangle.$$

Clearly no computation under A_3 exists. Starting with a word in which $\langle 1\rangle$ occurs, A_3 manufactures more and more one's.

Lemma 4.2. *Every Turing computable function is algorithmic.*

PROOF. Let f (n-ary) be computed by a Turing machine M, with notation as in 1.1 and 3.9. With each row $ti = (c_{j(i)}, \varepsilon i, vi, di)$ of M $(1 \leq i \leq 2m)$ we shall associate one or more rows $t'(i, 0), \ldots, t'(i, pi)$ of a Markov algorithm, depending on vi.

Case 1. $vi = 0$ or 1. We associate the row

$$\langle \varepsilon_i \quad 2 \quad 1^{(cji+1)} \quad 2\rangle \to \langle v_i \quad 2 \quad 1^{(di+1)} \quad 2\rangle.$$

Case 2. $vi = 2$. We associate the rows (in order)

$$\begin{aligned}
\langle 0 \quad \varepsilon_i \quad 2 \quad 1^{(cji+1)} \quad 2\rangle &\to \langle 0 \quad 2 \quad 1^{(di+1)} \quad 2 \quad \varepsilon_i\rangle \\
\langle 1 \quad \varepsilon_i \quad 2 \quad 1^{(cji+1)} \quad 2\rangle &\to \langle 1 \quad 2 \quad 1^{(di+1)} \quad 2 \quad \varepsilon_i\rangle \\
\langle \varepsilon_i \quad 2 \quad 1^{(cji+1)} \quad 2\rangle &\to \langle 0 \quad 2 \quad 1^{(di+1)} \quad 2 \quad \varepsilon_i\rangle
\end{aligned}$$

Case 3. $vi = 3$. We associate the rows (in order)

$$\begin{aligned}
\langle \varepsilon_i \quad 2 \quad 1^{(cji+1)} \quad 2 \quad 0\rangle &\to \langle \varepsilon_i \quad 0 \quad 2 \quad 1^{(di+1)} \quad 2\rangle \\
\langle \varepsilon_i \quad 2 \quad 1^{(cji+1)} \quad 2 \quad 1\rangle &\to \langle \varepsilon_i \quad 1 \quad 2 \quad 1^{(di+1)} \quad 2\rangle \\
\langle \varepsilon_i \quad 2 \quad 1^{(cji+1)} \quad 2\rangle &\to \langle \varepsilon_i \quad 0 \quad 2 \quad 1^{(di+1)} \quad 2\rangle.
\end{aligned}$$

Case 4. $vi = 4$. We associate the row

$$\langle \varepsilon_i \quad 2 \quad 1^{(cji+1)} \quad 2\rangle \to \cdot\langle \varepsilon_i \quad 2\rangle.$$

Now let A be the following Markov algorithm:

$$\begin{aligned}
&t'(1, 0) \\
&\quad\vdots \\
&t'(1, p1) \\
&t'(2, 0) \\
&\quad\vdots \\
&t'(2m, p(2m)) \\
&\langle 2\rangle \to \langle 2 \quad 1^{(c1+1)} \quad 2\rangle.
\end{aligned}$$

We claim that A computes f. To see this, let $x_0, \ldots, x_{n-1} \in \omega$. Since M computes f, by 3.9 there is a computation $\langle (F, c_1, 0), (G_1, a_1, b_1), \ldots, (G_{q-1}, a_{q-1}, b_{q-1})\rangle$ of M with the following properties:

(1) $0 \quad 1^{(x0+1)} \quad 0 \quad \cdots \quad 0 \quad 1^{(x(m-1)+1)}$ lies on F ending at -1, and F is 0 elsewhere;

(2) $1^{(x(m-1)+1)} \quad 0 \quad 1^{(f(x0,\ldots,x(m-1))+1)} \quad 0$ lies on G_{q-1} ending at b_{q-1}.

Now let $G_0 = F$, $a_0 = c_1$, $b_0 = 0$. Let Q_{-1} be the word

$$0 \quad 1^{(x0+1)} \quad 0 \quad \cdots \quad 0 \quad 1^{(x(m-1)+1)} \quad 0 \quad 2.$$

Now we define N_i, P_i, Q_i for $i < q$ by induction. Let N_0 be $0 \; 1^{(x0+1)} \; 0 \cdots 0 \; 1^{(x(m-1)+1)} \; 0$, $P_0 = 0$ (the empty sequence), and $Q_0 = 0 \; 1^{(x0+1)} \; 0 \cdots 0 \; 1^{(x(m-1)+1)} \; 0 \; 2 \; 1^{(c1+1)} \; 2$. Suppose now that $i + 1 < q$ and that N_i, P_i, Q_i have been defined so that the following conditions hold:

(3) $N_i \neq 0$;

(4) N_i lies on G_i ending at b_i;

(5) P_i lies on G_i beginning at $b_i + 1$;

(6) G_i is 0 except for $N_i P_i$;
(7) exactly two 2's occur in Q_i;
(8) $N_i\ 2\ 1^{(ai+1)}\ 2\ P_i = Q_i$;
(9) if $i \neq 0$, then (Q_{i-1}, Q_i) is a nonterminating algorithmic step under A.

Clearly (3)–(9) hold for $i = 0$. We now define $N_{i+1}, P_{i+1}, Q_{i+1}$. Let the row of M beginning with $a_i\ \ G_i b_i$ be

$$a_i \quad G_i b_i \quad v \quad w.$$

We now distinguish cases depending on v. Note that, since $i < q - 1$, $v \neq 4$. In each case we define $N_{i+1}, P_{i+1}, Q_{i+1}$, and it will then be evident that (3)–(9) hold for $i + 1$ in that case. In each case, let Q_{i+1} be defined by (8) for $i + 1$.

Case 1. $v = 0$. Let N_{i+1} be N_i with its last entry replaced by 0, and let $P_{i+1} = P_i$.

Case 2. $v = 1$. Similarly.

Case 3. $v = 2$. Here we take two subcases:

Subcase 1. N_i has length at least 2. Write $N_i = N_{i+1}\varepsilon$, where $\varepsilon = 0$ or 1, and set $P_{i+1} = \varepsilon P_i$.

Subcase 2. N_i has length 1. Let $N_{i+1} = \langle 0 \rangle$, $P_{i+1} = N_i P_i$.

Case 4. $v = 3$. Again we take two subcases:

Subcase 1. $P_i \neq 0$. Write $P_i = \varepsilon P_{i+1}$ with $\varepsilon = 0$ or 1, and set $N_{i+1} = N_i\varepsilon$.

Subcase 2. $P_i = 0$. Let $P_{i+1} = 0$, $N_{i+1} = N_i 0$.

This completes the definition of N_i, P_i, Q_i for all $i < q$, so that (3)–(9) hold. Let Q_q be the word $N_{q-1}\ 2$. Then by (9) it follows that $\langle Q_{-1}, Q_0, \ldots, Q_q \rangle$ is a computation under A. Now by (2), (6), and (4), we can write

$$N_{q-1} = N'_{q-1} \quad 0 \quad 1^{(f(x0,\ldots,x(m-1))+1)} \quad 0;$$

hence $0\ 1^{(f(x0,\ldots,x(m-1))+1)}\ 0\ 2$ occurs in Q_q and 2 occurs only once in Q_q.

It follows that A computes f. □

We now turn to the problem of showing that every algorithmic function is recursive. This is done by the now familiar device of Gödel numbering.

Definition 4.3. If $a = \langle a_0, \ldots, a_{m-1} \rangle$ is a word, its Gödel number, $\mathcal{g}a$, is

$$\prod_{i<m} p_i^{ai+1}.$$

Thus the empty word has Gödel number 1.

Lemma 4.4. *The set of Gödel numbers of words is elementary.*

PROOF. m is the Gödel number of a word iff $m = 1$ or $m > 1$ and $\forall i \leqslant lm$ $[(m)_i \leq 3$ and $1 \leq (m)_i]$. □

Definition 4.5. If A is a Markov algorithm as in 4.1, its Gödel number, gA, is the number

$$\prod_{i<m} p_i^{ti},$$

where $ti = 2^{gai} \cdot 3^{gbi} \cdot 5^{ci}$ for each $i \leq m$.

Lemma 4.6. *The set of Gödel numbers of Markov algorithms is elementary.*

Proof. n is the Gödel number of a Markov algorithm iff $n \geq 2$ and $\forall i \leq \mathrm{l}n$ $[((n)_i)_0$ and $((n)_i)_1$ are Gödel numbers of words, $((n)_i)_2 \leq 1$, and $\mathrm{l}((n)_i) \leq 2]$. □

Definition 4.7. Let $R_0 = \{(m, n)$: m and n are Gödel numbers of words a and b respectively, and a occurs in $b\}$.

Lemma 4.8. *R_0 is elementary.*

Proof $(m, n) \in R_0$ iff m is the Gödel number of a word, n is the Gödel number of a word, and $\exists x \leq n\, \exists y \leq n[\mathrm{Cat}\,(\mathrm{Cat}\,(x, m), y) = n]$. (Recall from 3.30 the definition of Cat.) □

Definition 4.9. $R_1 = \{(m, n, p, q)$: m, n, p, q are Gödel numbers of words a, b, c, d respectively, and (a, b, c) is the first occurrence of b in $d\}$.

Lemma 4.10. *R_1 is elementary.*

Proof. $(m, n, p, q) \in R_1$ if m, n, p, q are Gödel numbers of words and $\mathrm{Cat}\,(\mathrm{Cat}\,(m, n), p) = q$ and $\forall x \leq q\, \forall y \leq q[\mathrm{l}x < \mathrm{l}m\ \&\ x$ and y are Gödel numbers of words $\Rightarrow \mathrm{Cat}\,(\mathrm{Cat}\,(x, n), y) \neq q]$. □

Definition 4.11. $R_2 = \{(p, m, n) : p$ is the Gödel number of a Markov algorithm A, m, n are Gödel numbers of words a, b respectively, and (a, b) is a nonterminating computation step under $A\}$.

Lemma 4.12. *R_2 is elementary.*

Proof. $(p, m, n) \in R_2$ iff p is the Gödel number of a Markov algorithm, m and n are Gödel numbers of words, $\exists i \leq \mathrm{l}p$ such that $(((p)_i)_0, m) \in R_0$, and $\forall i \leq \mathrm{l}p\, \forall x \leq m\, \forall y \leq m[(((p)_i)_0, m) \in R_0\ \&\ \forall j < i[(((p)_j)_0, m) \notin R_0]\ \&\ (x, ((p)_i)_0, y, m) \in R_1 \Rightarrow \mathrm{Cat}\,(\mathrm{Cat}\,(x, ((p)_i)_1, y) = n\ \&\ ((p)_i)_2 = 0]$. □

Definition 4.13. R_3 is like R_2 except with "terminating" instead of "nonterminating".

Lemma 4.14. *R_3 is elementary.*

Definition 4.15. If $\langle d_0, \ldots, d_m \rangle$ is a finite sequence of words, its Gödel number is

$$\prod_{i<m} p_i^{\mathscr{g}d_i}.$$

Also let $R_4 = \{(m, n) : m$ is the Gödel number of a Markov algorithm A, and n is the Gödel number of a computation under $A\}$.

Lemma 4.16. *R_4 is elementary.*

PROOF. $(m, n) \in R_4$ iff m is a Gödel number of a Markov algorithm, $\mathrm{l}n \geq 1$, and $\forall i < \mathrm{l}n \mathbin{\dot-} 1[(m, (n)_i, (n)_{i+1}) \in R_2]$ and $(m, (n)_{\mathrm{l}n \dot- 1}, (n)_{\mathrm{l}n}) \in R_3$. □

Definition 4.17. $f_1 x = \prod_{i \leq x} p_i^2$.

Lemma 4.18. *f_1 is elementary.*

Definition 4.19. $f_2^1 x = \mathrm{Cat}\,(2, f_1 x) \cdot f_2^{m+1}(x_0, \ldots, x_m) = \mathrm{Cat}\,(f_2^m(x_0, \ldots, x_{m-1},)$ $f_2^1 x_m)$.

Lemma 4.20. *f_2^m is elementary, for each $m \in \omega \sim \{0\}$.*

Lemma 4.21. *$f_2^m(x_0, \ldots, x_{m-1})$ is the Gödel number of*

$$0 \quad 1^{(x0+1)} \quad 0 \quad \cdots \quad 0 \quad 1^{(x(m-1)+1)}.$$

The notations R_1, R_2, R_3, R_4, f_1, f_2^m will not be used beyond the present section.

Definition 4.22. $T'_m = \{(e, x_0, \ldots, x_{m-1}, c) : e$ is the Gödel number of a Markov algorithm A, and c is the Gödel number of a computation $\langle d_0, \ldots, d_n \rangle$ under A, $(c)_0 = \mathrm{Cat}\,(f_2^m(x_0, \ldots, x_{m-1}), 2 \cdot 3^3)$, and 2 occurs only once in $d_n\}$.

Lemma 4.23. *T'_m is elementary.*

Definition 4.24. $V'y = \mu x \leq y[(\mathrm{Cat}\,(f_2^1 x, 2 \cdot 3^3), (y)_{\mathrm{l}y}) \in R_0]$.

Lemma 4.25. *V' is elementary.*

Lemma 4.26. *Every algorithmic function is recursive.*

PROOF. Say f is m-ary and is computed by a Markov algorithm A. Let e be the Gödel number of A. Then for any $x_0, \ldots, x_{m-1} \in \omega$ we have

$$f(x_0, \ldots, x_{m-1}) = V'\mu z(\langle e, x_0, \ldots, x_{m-1}, z \rangle \in T'_m).$$

Thus f is recursive, as desired. □

Theorem 4.27. *Turing computable = recursive = algorithmic.*

BIBLIOGRAPHY

1. Curry, H. *Foundations of Mathematical Logic*. New York: McGraw-Hill (1963).
2. Detlovs, V. *The equivalence of normal algorithms and recursive functions*. A.M.S. Translations Ser. 2, Vol. 23, pp. 15–81.
3. Markov, A. *Theory of Algorithms*. Jerusalem: Israel Program for Scientific Translations (1961).

EXERCISES

4.28. Let A be the algorithm

$$\begin{array}{rl} 2\ 0 \rightarrow & 0\ 2 \\ 2\ 1 \rightarrow & 1\ 2 \\ 2 \rightarrow & \cdot\ \ 1^{(3)} \\ \rightarrow & 2 \end{array}$$

Show that A converts any word a on 0, 1 (i.e., involving only 0 and 1) into $a\ 1^{(3)}$.

4.29. Construct an algorithm which converts every word into a fixed word a.

4.30. Construct an algorithm which converts every word a into $1^{(n+1)}$, where n is the length of a.

4.31. Let a be a fixed word. Construct an algorithm which converts any word $\neq a$ into the empty word, but leaves a alone.

4.32. There is no algorithm which converts any word a into aa.

4.33. Construct an algorithm which converts any word a on 0, 1 into aa.

4.34*. Show directly that any algorithmic function is Turing computable.

5 Recursion Theory

We have been concerned so far with just the definitions of mathematical notions of effectiveness. We now want to give an introduction to the theory of effectiveness based on these definitions. Most of the technical details of the proofs of the results of this chapter are implicit in our earlier work. We wish to look at the proofs and results so far stated and try to see their significance.

In order to formulate some of the results in their proper degree of generality we need to discuss the notion of *partial functions*. An m-ary partial function on ω is a function f mapping some *subset* of ${}^m\omega$ into ω. The domain of f may be empty—then f itself is the empty set. The domain of f may be finite; it may also be infinite but not consist of all of ${}^m\omega$. Finally, it may be all of ${}^m\omega$, in which case f is an ordinary m-ary function on ω. When talking about partial functions, we shall sometimes refer to those f with $\operatorname{Dmn} f = {}^m\omega$ as *total*.

Intuitively speaking, a partial function f (say m-ary) is *effective* if there is an automatic procedure P such that for any $x_0, \ldots, x_{m-1} \in \omega$, if P is presented with the m-tuple $\langle x_0, \ldots, x_{m-1}\rangle$ then it proceeds to calculate, and if $\langle x_0, \ldots, x_{m-1}\rangle \in \operatorname{Dmn} f$, then after finitely many steps P produces the answer $f(x_0, \ldots, x_{m-1})$ and stops. In case $(x_0, \ldots, x_{m-1}) \notin \operatorname{Dmn} f$ the procedure P never stops. We do *not* require that there be an automatic method for recognizing membership in $\operatorname{Dmn} f$. Clearly if f is total then this notion of effectiveness coincides with our original intuitive notion (see p. 12). Now we want to give mathematical equivalents for the notion of an effective partial function.

Definition 5.1. Let f be an m-ary partial function. We say that f is *partial Turing computable* iff there is a Turing machine M as in 1.1 such that for every tape description F, all $q, n \in \mathbb{Z}$, and all $x_0, \ldots, x_{m-1} \in \omega$, if $0\ 1^{(x0+1)}\ 0 \cdots 0\ 1^{(x(m-1)+1)}$ lies on F beginning at q and ending at n, and if $Fi = 0$ for all $i > n$, then the two conditions (*i*) $(x_0, \ldots, x_{m-1}) \in \mathrm{Dmn}\, F$,

(*ii*) there is a computation of M beginning with $(F, c_1, n + 1)$

are equivalent; and if one of them holds, and $\langle(F, c_1, n + 1), (G_1, a_1, b_1), \ldots, (G_{p-1}, a_{p-1}, b_{p-1})\rangle$ is a computation of M, then (1)–(3) of 3.9(*ii*) hold.

Clearly any partial Turing computable function is effectively calculable.

Corollary 5.2. *Every Turing computable function is partial Turing computable. Every total partial Turing computable function is Turing computable.*

Next, we want to generalize our Definition 3.1 of recursive functions. To shorten some of our following exposition we shall use the informal notation

$$\cdots \simeq ---$$

to mean that $\cdots$ is defined iff $---$ is defined, and if $\cdots$ is defined, then $\cdots = ---$. For example, if f is the function with domain $\{2, 3\}$ then when we say

$$gx + hx \simeq f(x + 2) \qquad \text{for all } x \in \omega,$$

we mean that $\mathrm{Dmn}\, g \cap \mathrm{Dmn}\, h = \{0,1\}$ and for any $x \in \{0,1\}$, $gx + hx = f(x + 2)$.

Definition 5.3

(*i*) *Composition.* We extend the operator K^m_n of 2.1 to act upon partial functions. Let f be an m-ary partial function, and $g_0, \ldots, g_{m-1}$ n-ary partial functions. Then K^m_n is the n-ary partial function h such that for any $x_0, \ldots, x_{n-1} \in \omega$,

$$h(x_0, \ldots, x_{n-1}) \simeq f(g_0(x_0, \ldots, x_{n-1}), \ldots, g_{m-1}(x_0, \ldots, x_{n-1})).$$

(*ii*) *Primitive recursion with parameters.* If f is an m-ary partial function and h is an $(m + 2)$-ary partial function, then $\mathrm{R}^m(f, h)$ is the $(m + 1)$-ary partial function defined recursively by:

$$g(x_0, \ldots, x_{m-1}, 0) \simeq f(x_0, \ldots, x_{m-1})$$
$$g(x_0, \ldots, x_{m-1}, \mathrm{s}y) \simeq h(x_0, \ldots, x_{m-1}, y, g(x_0, \ldots, x_{m-1}, y))$$

for all $x_0, \ldots, x_{m-1}, y \in \omega$.

(*iii*) *Primitive recursion without parameters.* If $a \in \omega$ and h is a 2-ary partial function, then $\mathrm{R}^0(a, h)$ is the unary partial function g defined recursively by

$$g0 = a$$
$$g\mathrm{s}y \simeq h(y, gy)$$

for all $y \in \omega$.

(*iv*) *Minimalization*. Let f be an $(m + 1)$-ary partial function. An m-ary partial function g is obtained from f by *minimalization* provided that for all $x_0, \ldots, x_{m-1} \in \omega$,

$$g(x_0, \ldots, x_{m-1}) \simeq \text{least } y \text{ such that } \forall z \leq y((x_0, \ldots, x_{m-1}, z) \in \mathrm{Dmn}\, f) \text{ and } f(x_0, \ldots, x_{m-1}, y) = 0.$$

We then write $g(x_0, \ldots, x_{m-1}) \simeq \mu y(f(x_0, \ldots, x_{m-1}, y) = 0)$.

(*v*) The class of *partial recursive functions* is the intersection of all classes C of partial functions such that $\mathscr{s} \in C$, $U_i^n \in C$ whenever $i < n \in \omega$, and C is closed under composition, primitive recursion, and minimalization.

Clearly every partial recursive function is effectively calculable. Note that it is not appropriate to simplify the definition of minimalization to

$$g(x_0, \ldots, x_{m-1}) \simeq \text{least } y \text{ such that } (x_0, \ldots, x_{m-1}, y) \in \mathrm{Dmn}\, f \text{ and } f(x_0, \ldots, x_{m-1}, y) = 0,$$

for all $x_0, \ldots, x_{m-1} \in \omega$. For, even if f is calculable there may be no clear way to calculate g. For example, suppose that $(x_0, \ldots, x_{m-1}, 0) \notin \mathrm{Dmn}\, f$, while $(x_0, \ldots, x_{m-1}, 1) \in \mathrm{Dmn}\, f$ and $f(x_0, \ldots, x_{m-1}, 1) = 0$. Without knowing that $(x_0, \ldots, x_{m-1}, 0) \notin \mathrm{Dmn}\, f$ it is unclear at what point in a computation of $g(x_0, \ldots, x_{m-1})$ one would be justified in setting $g(x_0, \ldots, x_{m-1}) = 1$. The above definition of minimalization clearly avoids this difficulty. One can give explicit examples where f is partial recursive but g, defined in this new way, is not. (See Exercise 5.38.)

Note that there *are* nontotal partial recursive functions. For example, clearly C_1^2 is partial recursive, and hence by 5.3(*iv*) so is the function g such that $gx \simeq \mu y(C_1^2(x, y) = 0)$. Obviously, however, g is the empty function.

Corollary 5.4. *Every general recursive function is partial recursive.*

In contrast to the situation for Turing computability, it is not at all immediately clear that every total partial recursive function is general recursive; this is, however, true, as our next theorem shows. The proof of this theorem is rather long when carried out from the beginning.

Theorem 5.5. *Partial Turing computable = partial recursive.*

PROOF. PARTIAL RECURSIVE ⇒ PARTIAL TURING COMPUTABLE. Here it is only necessary to read again the proofs of Lemmas 3.10–3.16 and check that they adapt to the situation of partial functions and the new Definitions 5.1 and 5.3.

PARTIAL TURING COMPUTABLE ⇒ PARTIAL RECURSIVE. Again one needs only to reread 3.17–3.38. □

Corollary 5.6. *Any total partial recursive function is recursive.*

A natural question occurs as to whether every partial recursive function can be extended to a recursive function; the answer is no:

Theorem 5.7. *There is a partial recursive function f such that f cannot be extended to a recursive function.*

PROOF. The rule for computing f is as follows. For a given $x \in \omega$, determine whether or not x is the Gödel number of a Turing machine. If it is not, set $fx = 0$. If it is, test in succession whether or not $(x, x, 0) \in T_1$, $(x, x, 1) \in T_1$, $(x, x, 2) \in T_1$, etc. The first time we find a u such that $(x, x, u) \in T_1$, set $fx = Vu + 1$. If we never find such a u, the computation never ends. Clearly f is intuitively a calculable partial function, and it has the following property: for any $x \in \omega$,

$$(1)\quad \begin{aligned} &fx = 0 && \text{if } x \text{ is not the Gödel number of a Turing machine,} \\ & && V\mu u((x, x, u) \in T_1) + 1 \text{ if } x \text{ is the Gödel number} \\ & && \text{of a Turing machine and there is such a } u, \\ &fx \text{ is undefined, otherwise.} \end{aligned}$$

It is routine to show that f is partial recursive; we will prove this formally in this case, but usually not in the future. We can *define* f by (1). Clearly then, for any $x \in \omega$

$$fx \simeq [V\mu u(\chi_{T_1}(x, x, u) = 1 \text{ or } \chi_T x = 0) + 1] \cdot \chi_T x,$$

so f is partial recursive.

Now f cannot be extended to a general recursive function. For, suppose $f \subseteq h$ with h general recursive. By the proof of 3.38 there is an $e \in \omega$ such that, for all $x \in \omega$,

$$hx = V\mu u(e, x, u) \in \mathrm{T}_1).$$

In particular, $he = V\mu u((e, e, u) \in \mathrm{T}_1)$ and (by the definition of T_1, 3.35) e is the Gödel number of a Turing machine. Thus fe is defined, and

$$fe = V\mu u((e, e, u) \in \mathrm{T}_1) + 1 = he + 1 = fe + 1,$$

contradiction. □

We now turn to the formulation of some basic results called the *normal form*, *iteration*, and *recursion theorems*.

Definition 5.8. For any $e \in \omega$ and $m \in \omega$ let φ_e^m be the m-ary partial recursive function such that for all $x_0, \ldots, x_{m-1} \in \omega$,

$$\varphi_e^m(x_0, \ldots, x_{m-1}) \simeq V\mu u((e, x_0, \ldots, x_{m-1}, u) \in \mathrm{T}_m).$$

Note also that the $(m + 1)$-ary partial function φ' defined by

$$\varphi'(x_0, \ldots, x_{m-1}, e) \simeq V\mu u((e, x_0, \ldots, x_{m-1}, u) \in \mathrm{T}_m)$$

for all $x_0, \ldots, x_{m-1}, e \in \omega$, is also partial recursive. This remark will be frequently useful in what follows.

Theorem 5.9 (Normal form theorem). *For any partial recursive function f (say m-ary) there is an $e \in \omega$ such that $f = \varphi_e^m$.*

Proof. By the proof of 5.5, second part. □

This theorem, which was implicitly used already in the proof of 5.7, has many important corollaries, which we shall now explore. First of all, its normal form nature is made a little more explicit in the following corollary.

Corollary 5.10. *For each $m \in \omega \sim 1$ there exist a 1-place elementary function f and an $(m+2)$-place elementary function g such that for any m-ary partial recursive function h there is an $e \in \omega$ such that for all $x_0, \ldots, x_{m-1} \in \omega$,*

$$h(x_0, \ldots, x_{m-1}) \simeq f\mu u[g(e, x_0, \ldots, x_{m-1}, u) = 0].$$

Proof. Let $f = \mathrm{V}$ and $g = \overline{\mathrm{sg}} \circ \chi_{\mathrm{T}m}$. □

This formulation suggests the possibility of improving the result by dropping f. (Another possibility, dropping μ, is impossible since there are recursive functions which are not primitive recursive.) As to this possibility, see Exercise 5.43; the answer is no.

Theorem 5.9 and its proof give rise to a certain *universal phenomenon* as follows.

Corollary 5.11 (Universal Turing machines). *There is a Turing machine M with the following property. If f is any unary partial Turing computable function and a Turing machine N computes it, and if e is the Gödel number of N, then if $0\ 1^{(e+1)}\ 0\ 1^{(x+1)}\ 0$ is placed upon an otherwise blank tape ending at -1 and if M is started at 0, then M will stop iff $x \in \mathrm{Dmn}\, f$, and if $x \in \mathrm{Dmn}\, f$, then after the machine stops $1^{(fx+1)}\ 0$ will lie on the tape beginning at 1.*

Proof. Let g be the partial recursive function defined by

$$g(e, x) \simeq \mathrm{V}\mu u[(e, x, u) \in \mathrm{T}_1]$$

for all $e, x \in \omega$. Let M compute g. Clearly M is as desired. □

In more intuitive terms we can describe the way M is to act as follows: M is presented with two numbers e and x. First M checks if e is the Gödel number of some Turing machine. If it is, say $e = \mathscr{g}N$, then M begins checking one after the other whether 0 or 1 or $\cdots$ is the Gödel number of a computation under N with input x. If there is such a number, M takes the first such and reads off the result of the computation. It may be that e is not the Gödel number of a Turing machine or that there is no computation with input x; then M does not give an answer.

Corollary 5.12 (Universal partial recursive function). *There is a partial recursive function g of two variables such that for any partial recursive function f of one variable there is an $e \in \omega$ such that for all $x \in \omega$, $g(e, x) \simeq fx$.*

PROOF. Let g be as in the proof of 5.11. □

In view of the proof of 3.4, the reader might view 5.12 with some suspicion. Let us see what happens if we try the diagonal method on the g of 5.12. For any $x \in \omega$, let $fx \simeq g(x, x) + 1$. Then f is partial recursive, so by 5.12 there is an $e \in \omega$ such that for all $x \in \omega$, $g(e, x) \simeq fx$. Now if $g(e, e)$ is defined, then $g(e, e) = fe = g(e, e) + 1$. Conclusion: $g(e, e)$ is not defined. We are saved by g being a partial function. No contradiction arises.

We now turn to the iteration theorem. This basic result, although of a rather technical nature, is basic for most of the deeper results in recursion theory. See, e.g., the proofs of 5.15, 6.19, and 6.25.

Theorem 5.13 (Iteration theorem). *For any $m, n \in \omega \sim 1$ there is an $(m + 1)$-ary recursive function s_n^m such that for all $e, y_1, \ldots, y_m, x_1, \ldots, x_n \in \omega$,*

$$\varphi_e^{m+n}(x_1, \ldots, x_n, y_1, \ldots, y_m) \simeq (\varphi^n(\mathrm{s}_n^m(e, y_1, \ldots, y_m)))(x_1, \ldots, x_n).$$

PROOF. If M is any Turing machine and $y_1, \ldots, y_m \in \omega$, let $M^*_{y1,\ldots,ym}$ be the following Turing machine:

$$\begin{aligned}\text{Start} &\to (T_{\text{left}} \to T_1)^{y1} \to (T_{\text{left}} \to T_1)^{y1} \to \cdots \\ &\to (T_{\text{left}} \to T_1)^{ym} \to T_{\text{left}} \to M \to T^m_{\text{lshift}} \to \text{Stop}\end{aligned}$$

Clearly there is an $(m + 1)$-ary recursive function s_n^m such that for any $e, y_1, \ldots, y_m \in \omega$, if e is the Gödel number of a Turing machine M, then $\mathrm{s}_n^m(e, y_1, \ldots, y_m)$ is the Gödel number of $M^*_{y1,\ldots,ym}$. Obviously s_n^m is as desired in the theorem. □

Actually a more detailed analysis would show that s_n^m in 5.13 can be taken to be elementary recursive, but we shall not use this fact. As a first application of the iteration theorem we give

Corollary 5.14. *There is no binary function f such that for all $x, y \in \omega$,*

$$\begin{aligned} f(x, y) &= 1 \qquad \text{if } y \in \operatorname{Dmn} \varphi_x^1, \\ f(x, y) &= 0 \qquad \text{if } y \notin \operatorname{Dmn} \varphi_x^1. \end{aligned}$$

PROOF. Suppose there is such an f; say $f = \varphi_e^2$. Now for any $x, y \in \omega$ let

$$g(x, y) \simeq \mu z[\mathrm{V}\mu u((y, x, x, u) \in \mathrm{T}_2) = 0].$$

Hence for any $x, y \in \omega$,

$g(x, y) = 0$ if y is the Gödel number of a Turing machine, $(x, x) \in \operatorname{Dmn} \varphi_y^2$, and $\varphi_y^2(x, x) = 0$;
$g(x, y)$ is undefined, otherwise.

Say $g = \boldsymbol{\varphi}_r^2$. Then by the iteration theorem,

$$\begin{aligned} s_1^1(r, e) \in \text{Dmn}\, \varphi'(s_1^1(r, e)) \quad & \text{iff } (s_1^1(r, e), e) \in \text{Dmn}\, \boldsymbol{\varphi}_r^2 \\ & \text{iff } f(s_1^1(r, e), s_1^1(r, e)) = 0 \\ & \text{iff } s_1^1(r, e) \notin \text{Dmn}\, \boldsymbol{\varphi}'(s_1^1(r, e)), \end{aligned}$$

a contradiction. □

Thus there is no automatic method for determining of a pair (x, y) whether $y \in \text{Dmn}\, \boldsymbol{\varphi}_x^1$. Otherwise stated, there is no automatic method of determining of a Turing machine M and a number y whether M will eventually stop with an output when presented with input y. Thus Corollary 5.14 shows the *recursive unsolvability of the Halting problem for Turing machines*. We can give a more intuitive, informal proof of this result. Suppose we have an automatic method telling us whether a Turing machine M will stop with input y. Then we can construct a machine N such that for any $y \in \omega$ the following conditions are equivalent:

(1) N stops when given input y;
(2) y is not the Gödel number of a Turing machine, or it is the number of a machine T such that T does not stop when given input y.

Let N have Gödel number e. By (1) and (2) we reach a contradiction in trying to decide whether N stops, given input e.

Theorem 5.15 (Recursion theorem). *If $m > 1$ and f is an m-ary partial recursive function, then there is an $e \in \omega$ such that for all $x_0, \ldots, x_{m-2} \in \omega$,*

$$f(x_0, \ldots, x_{m-2}, e) \simeq \boldsymbol{\varphi}_e^{m-1}(x_0, \ldots, x_{m-2}).$$

PROOF. For any $x_0, \ldots, x_{m-1} \in \omega$ let

$$g(x_0, \ldots, x_{m-1}) \simeq f(x_0, \ldots, x_{m-2}, s_{n-1}^1(x_{m-1}, x_{m-1})).$$

Thus g is partial recursive; say $g = \varphi_r^m$. Let $e = s_{m-1}^1(r, r)$. Then by the iteration theorem, for all $x_0, \ldots, x_{m-2} \in \omega$,

$$\begin{aligned} \boldsymbol{\varphi}_e^{m-1}(x_0, \ldots, x_{m-2}) & \simeq \boldsymbol{\varphi}_r^m(x_0, \ldots, x_{m-2}, r) \\ & \simeq g(x_0, \ldots, x_{m-2}, r) \\ & \simeq f(x_0, \ldots, x_{m-2}, e). \end{aligned}$$

□

The recursion theorem is extremely useful in checking that functions defined by rather complicated recursive conditions are, in fact, general recursive. We shall illustrate its use by verifying again that the functions of 3.5 and 3.52 are recursive.

In the case of 3.5, we first define an auxiliary three-place function h' that is obviously partial recursive from the form of its definition, which goes by cases as in the definition of h, as follows. Let $x, y, e \in \omega$.

Case 1. $x = 2$. Let $h'(x, y, e) = (y)_0 + 1$ for all y.

Case 2. $x = 2^n \cdot 3^{i+1}$, where $i < n$. Let $h'(x, y, e) = (y)_i$ for all y.

Case 3. $x = 2^n \cdot 5^n \cdot \mathrm{p}_3^q \cdot \mathrm{p}_4^{r0} \cdot \ldots \cdot \mathrm{p}_{m+3}^{r(m-1)}$, with $n, m > 0$.
Let

$$h'(x, y, e) \simeq \varphi_e^2(q, \mathrm{p}_0^{t0} \cdot \ldots \cdot \mathrm{p}_{m-1}^{t(m-1)}),$$

where $ti \simeq \varphi_e^2(ri, y)$ for each $i < m$.

Case 4. $x = 2 \cdot 7^q \cdot 11^a$ with $q > 0$. Let

$$h'(x, 1, e) = a,$$
$$h'(x, 2^{y+1}, e) \simeq \varphi_e^2(q, 2^y \cdot 3^v),$$

where $v \simeq \varphi_e^2(x, 2^y)$,

$$h'(x, z, e) = 0$$

for z not of the form 2^u.

Case 5. $x = 2^{m+1} \cdot 11^q \cdot 13^r$ with $m > 0$ and $q > 0$. Let y be given with $(y)_m = 0$. We set

$$h'(x, y, e) \simeq \varphi_e^2(q, y),$$
$$h'(x, y \cdot \mathrm{p}_m^{z+1}, e) \simeq \varphi_e^2(r, y \cdot \mathrm{p}_m^z \cdot \mathrm{p}_{m+1}^v),$$

where $v \simeq \varphi_e^2(x, y \cdot \mathrm{p}_m^z)$.

Case 6. For x not of one of the above forms, let $h'(x, y, e) = 0$ for all y, e.

Now we apply the recursion theorem and obtain an $e \in \omega$ such that for all $x, y \in \omega$,

$$h'(x, y, e) \simeq \varphi_e^2(x, y).$$

Now it is straightforward to check by complete induction on x that for all $x, y \in \omega$, $h(x, y) \simeq \varphi_e^2(x, y)$. Thus $h = \varphi_e^2$ is recursive.

It is similarly shown that the function in 3.52 is recursive. Namely, we define a partial recursive function f' as follows:

$$f'(0, y, e) = y + 1,$$
$$f'(x + 1, 0, e) \simeq \varphi_e^2(x, 1),$$
$$f'(x + 1, y + 1, e) \simeq \varphi_e^2(x, \varphi_e^2(x + 1, y)).$$

Let $e \in \omega$ be such that $f'(x, y, e) \simeq \varphi_e^2(x, y)$ for all $x, y \in \omega$. Then it is easily proved by induction on x, with induction on y in the induction step, that $f(x, y) \simeq \varphi_e^2(x, y)$ for all $x, y \in \omega$. Thus $f = \varphi_e^2$.

Theorem 5.16 (Fixed point theorem). *If f is a unary recursive function then there is an $e \in \omega$ such that $\varphi_e^1 = \varphi_{fe}^1$.*

Proof. For any $x, y \in \omega$ let

$$g(x, y) \simeq \mathrm{V}\mu u((fy, x, u) \in \mathrm{T}_1).$$

Thus g is partial recursive, and $g(x, y) \simeq \varphi_{fy}^1 x$ for all $x, y \in \omega$. Now we apply the recursion theorem to obtain on $e \in \omega$ such that $g(x, e) \simeq \varphi_e^1 x$ for all $x \in \omega$. Thus $\varphi_e^1 = \varphi_{fe}^1$, as desired. $\square$

An important consequence of the fixed-point theorem is given in

Theorem 5.17 (Rice). *Let F be a set of one-place partial recursive functions such that $0 \neq F$ and F does not consist of all one-place partial recursive functions. Then $A = \{e : \varphi_e^1 \in F\}$ is not recursive.*

PROOF. Suppose it is. Let $a \in A$ and $b \notin A$. Now define

$$\begin{aligned} gx &= a \qquad x \notin A, \\ gx &= b \qquad x \in A. \end{aligned}$$

Then g is recursive. By 5.16 choose e such that $\varphi_e^1 = \varphi_{ge}^1$. Then if $e \in A$ we see that $\varphi_e^1 \in F$ (by the definition of A), hence $\varphi_{ge}^1 \in F$, so $ge \in A$; but $e \in A$ implies also $ge = b \notin A$, contradiction. Also, $e \notin A$ implies on the one hand $\varphi_e^1 \notin F$, $\varphi_{ge}^1 \notin F$, $ge \notin A$, and on the other hand implies $ge = a \in A$, contradiction. □

Rice's theorem has many important corollaries; we shall mention a few.

Corollary 5.18. *For any unary partial recursive function f, $\{e : \varphi_e^1 = f\}$ is not recursive.*

Corollary 5.19. *$\{x : \varphi_x^1$ is a constant function$\}$ is not recursive.*

Corollary 5.20. *$\{(x, y) : y$ is in the range of $\varphi_x^1\}$ is not recursive.*

PROOF. If the given set is recursive, then clearly so is

$$\{x : 0 \text{ is in the range of } \varphi_x^1\},$$

contradicting 5.17. □

Corollary 5.21. *$\{(x, y) : \varphi_x^1 = \varphi_y^1\}$ is not recursive.*

PROOF. If the given set is recursive, and $e \in \omega$, then

$$\{x : \varphi_x^1 = \varphi_e^1\}$$

is recursive, contradicting 5.17. □

Thus there is no automatic procedure for determining whether or not φ_e^1 is a given unary partial recursive function; or whether or not φ_x^1 is a constant function; or whether or not y is in the range of φ_x^1; or whether or not $\varphi_x^1 = \varphi_y^1$. Clearly 5.14 is also a consequence of Rice's theorem.

We can use 5.14 to establish the following result concerning the length of computations.

Theorem 5.22. *There is no binary recursive function f such that for all e, $x \in \omega$, $\exists u((e, x, u) \in \mathrm{T}_1)$ iff $\exists u \leq f(e, x)((e, x, u) \in \mathrm{T}_1)$.*

PROOF. Suppose there is such an f. Let

$$g(e, x) = 1 \quad \text{if } \exists u \leq f(e, x)((e, x, u) \in \mathrm{T}_1),$$
$$g(e, x) = 0 \quad \text{otherwise.}$$

Thus g is recursive and

$$g(e, x) = 1 \quad \text{if } x \in \mathrm{Dmn}\, \varphi_e^1,$$
$$g(e, x) = 0 \quad \text{if } x \notin \mathrm{Dmn}\, \varphi_e^1,$$

contradicting 5.14. □

Thus there is no automatic procedure P such that, given a Turing machine M and a number x, P determines the maximum number of steps in an M-computation starting with x.

Our final topic of this section is the *arithmetical hierarchy*. The recursive relations, as we have argued, coincide with the effective number-theoretic relations. Certain other relations, namely those obtained by using the quantifiers $\exists$ or $\forall$ on the recursive relations, are also very natural relations to consider in many contexts. They can be arranged in the so-called arithmetical hierarchy, according to the depth of quantifiers used in defining them. We shall describe this hierarchy and its most important properties.

Our main result, 5.36, depends on the following normal form theorem.

Theorem 5.23. *Let $m > 1$. If R is an m-ary recursive relation, then there exist $e, e' \in \omega$ such that for all $x_0, \ldots, x_{m-2} \in \omega$,*

(i) $\exists y((x_0, \ldots, x_{m-2}, y) \in R)$ *iff* $\exists y((e, x_0, \ldots, x_{m-2}, y) \in \mathrm{T}_{m-1})$.

(ii) $\forall y((x_0, \ldots, x_{m-2}, y) \in R)$ *iff* $\forall y((e', x_0, \ldots, x_{m-2}, y) \notin \mathrm{T}_{m-1})$.

PROOF. For any $x_0, \ldots, x_{m-2} \in \omega$ let

$$f(x_0, \ldots, x_{m-2}) \simeq \mu y((x_0, \ldots, x_{m-2}, y) \in R).$$

Thus f is partial recursive, so by 5.9 there is an $e \in \omega$ such that for all $x_0, \ldots, x_{m-2} \in \omega$,

$$f(x_0, \ldots, x_{m-2}) \simeq \mathrm{V}\mu u((e, x_0, \ldots, x_{m-2}, u) \in \mathrm{T}_{m-1}).$$

Thus

$$\begin{aligned}\exists y((x_0, \ldots, x_{m-2}, y) \in R) \quad &\text{iff } (x_0, \ldots, x_{m-2}) \in \mathrm{Dmn}\, f,\\ &\text{iff } \exists y((e, x_0, \ldots, x_{m-2}, y) \in \mathrm{T}_{m-1}).\end{aligned}$$

Thus *(i)* holds. Condition *(ii)* is easily obtained from *(i)*. □

Definition 5.24. Let $\Sigma_0 = \Pi_0 =$ set of all recursive relations. If $n, m > 0$, then an n-ary relation R is in Σ_m (respectively Π_m) provided there is an $(m + n)$-ary recursive relation S such that, if m is odd,

$$\begin{aligned}R = \{(x_0, \ldots, x_{n-1}) \in {}^n\omega : \exists y_0 \in \omega\, \forall y_1 \in \omega\, \exists y_2 \in \omega \cdots\\ \forall y_{m-2} \in \omega\, \exists y_{m-1} \in \omega[(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}) \in S]\}\end{aligned}$$

(respectively

$$\begin{aligned}R = \{(x_0, \ldots, x_{n-1}) \in {}^n\omega : \forall y_0 \in \omega\, \exists y_1 \in \omega\, \forall y_2 \in \omega \cdots\\ \exists y_{m-2} \in \omega\, \forall y_{m-1} \in \omega[(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}) \in S]\}),\end{aligned}$$

while, if m is even,

$$R = \{(x_0, \ldots, x_{m-1}) \in {}^n\omega : \exists y_0 \in \omega \, \forall y_1 \in \omega \, \exists y_2 \in \omega \cdots \exists y_{m-2} \in \omega \, \forall y_{m-1} \in \omega[(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}) \in S]\}$$

(respectively

$$R = \{(x_0, \ldots, x_{n-1}) \in {}^n\omega : \forall y_0 \in \omega \, \exists y_1 \in \omega \, \forall y_2 \in \omega \cdots \forall y_{m-2} \in \omega \, \exists y_{m-1} \in \omega[(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}) \in S]\}).$$

Members of Σ_m (respectively Π_m) are called Σ_m-*relations* (respectively Π_m-*relations*). Also let $\Delta_m = \Sigma_m \cap \Pi_m$. Any member of $\bigcup_{m \in \omega \sim 1} (\Sigma_m \cup \Pi_m)$ is said to be *arithmetical.*

Note that there are only $\aleph_0$ arithmetic relations, and hence most number-theoretic relations are not arithmetical. Now we want to describe the relationships between the various classes Σ_m, Π_m, and indicate some operations under which these classes are closed. The following obvious proposition indicates how these classes can be inductively defined, and furnishes a basis for inductive proofs of our further results.

Proposition 5.25

(*i*) *An n-ary relation R is in* Σ_{m+1} *iff there is an* $(n + 1)$*-ary relation S in* Π_m *such that for all* $x_0, \ldots, x_{n-1} \in \omega$, $(x_0, \ldots, x_{n-1}) \in R$ *iff* $\exists y \in \omega ((x_0, \ldots, x_{n-1}, y) \in S)$.

(*ii*) *An n-ary relation R is in* Π_{m+1} *iff there is an* $(n + 1)$*-ary relation S in* Σ_m *such that for all* $x_0, \ldots, x_{n-1} \in \omega$, $(x_0, \ldots, x_{n-1}) \in R$ *iff* $\forall y \in \omega ((x_0, \ldots, x_{n-1}, y) \in S)$.

The following three propositions are now easily established by induction, using 5.25:

Proposition 5.26. *If R is an n-ary* Σ_m*-relation,* $f_0, \ldots, f_{n-1}$ *are n-ary recursive functions, and*

$$S = \{(x_0, \ldots, x_{n-1}) : (f_0(x_0, \ldots, x_{n-1}), \ldots, f_{n-1}(x_0, \ldots, x_{n-1})) \in R\},$$

then $S \in \Sigma_m$. *Similarly for* Π_m *and* Δ_m.

Proposition 5.27 (Adjunction of apparent variables). *If R is an n-ary* Σ_m*-relation and* $S = \{(x_0, \ldots, x_n) : (x_1, \ldots, x_n) \in R\}$, *then* $S \in \Sigma_m$. *Similarly for* Π_m *and* Δ_m.

Proposition 5.28 (Identification of variables). *If R is an n-ary* Σ_m*-relation,* $n > 1$, *and* $S = \{(x_0, \ldots, x_{n-2}) : (x_0, x_0, x_1, x_2, \ldots, x_{n-2}) \in R\}$, *then* $R \in \Sigma_m$. *Similarly for* Π_m *and* Δ_m.

Proposition 5.29. *If R and S are n-ary* Σ_m*-relations, then so are* $R \cup S$ *and* $R \cap S$. *Similarly for* Π_m *and* Δ_m.

PROOF. The assertions for Δ_m follow from those for Σ_m and Π_m. The assertions for Σ_m and Π_m are proved simultaneously by induction on m. The case $m = 0$ is trivial. Now assume the assertions for m. We take just one typical assertion for $m + 1$:

Assume $R, S \in \Sigma_{m+1}$; we show that $R \cap S \in \Sigma_{m+1}$. By 5.25, choose $(n + 1)$-ary Π_m-relations R', S' such that for all $x_0, \ldots, x_{n-1} \in \omega$,

$$\begin{aligned} (x_0, \ldots, x_{n-1}) \in R \quad & \text{iff } \exists y \in \omega[(x_0, \ldots, x_{n-1}, y) \in R'], \\ (x_0, \ldots, x_{n-1}) \in S \quad & \text{iff } \exists y \in \omega[(x_0, \ldots, x_{n-1}, y) \in S']. \end{aligned}$$

Then for any $x_0, \ldots, x_{n-1} \in \omega$,

$$\begin{aligned} (x_0, \ldots, x_{n-1}) \in R \cap S \quad & \text{iff } \exists y \in \omega\, \exists z \in \omega[(x_0, \ldots, x_{n-1}, y) \in R' \\ & \qquad \text{and } (x_0, \ldots, x_{n-1}, z) \in S']. \end{aligned}$$

Now let $R'' = \{(x_0, \ldots, x_{n+1}) : (x_0, \ldots, x_n) \in R'\}$ and $S'' = \{(x_0, \ldots, x_{n+1}) : (x_0, \ldots, x_{n-1}, x_{n+1}) \in S'\}$. Using 5.26 and 5.27 it is easy to see that R'', $S'' \in \Pi_m$. Now, continuing from above, for any $x_0, \ldots, x_{n-1} \in \omega$,

$$\begin{aligned} (x_0, \ldots, x_{n-1}) \in R \cap S \quad & \text{iff } \exists y \in \omega\, \exists z \in \omega[(x_0, \ldots, x_{n-1}, y, z) \in R'' \cap S''] \\ & \text{iff } \exists y \in \omega[(x_0, \ldots, x_{n-1}, (y)_0, (y)_1) \in R'' \cap S'']. \end{aligned}$$

Since $R'' \cap S'' \in \Pi_m$ by the induction hypothesis, we get $R \cap S \in \Sigma_{m+1}$ by 5.25. □

For $m > 0$, neither Σ_m nor Π_m nor Δ_m is closed under complementation; see 5.36. The following proposition is evident.

Proposition 5.30. *If R is an n-ary Σ_m-relation with $n > 1$ and $m > 0$, and if $S = \{(x_0, \ldots, x_{n-2}) : \exists y \in \omega(x_0, \ldots, x_{n-2}, y) \in R\}$, then $S \in \Sigma_m$. Similarly with Π_m and $\forall$.*

Proposition 5.31. *If R is an n-ary Σ_m-relation, then so are the two relations*

$$\begin{aligned} S &= \{(x_0, \ldots, x_{n-1}) : \exists y < x_{n-1}[(x_0, \ldots, x_{n-2}, y) \in R], \\ T &= \{(x_0, \ldots, x_{n-1}) : \forall y < x_{n-1}[(x_0, \ldots, x_{n-2}, y) \in R]. \end{aligned}$$

Similarly for Π_m and Δ_m.

PROOF. Again we prove all cases simultaneously by induction on m. The case $m = 0$ is trivial. Assume that all of the statements are true for m. We take one typical case for $m + 1$:

Let R be an n-ary Σ_{m+1}-relation, and let T be as above. By 5.25, let R' be a Π_m-relation such that for all $x_0, \ldots, x_{n-1} \in \omega$,

$$(x_0, \ldots, x_{n-1}) \in R \quad \text{iff } \exists z \in \omega[(x_0, \ldots, x_{n-1}, z) \in R'].$$

Clearly, then, it suffices to show that for all $x_0, \ldots, x_{n-1} \in \omega$,

$$\begin{aligned} (1) \quad & \forall y < x_{n-1}\, \exists z \in \omega[(x_0, \ldots, x_{n-2}, y, z) \in R'] \\ & \qquad \text{iff } \exists z \in \omega\, \forall y < x_{n-1}[(x_0, \ldots, x_{n-2}, y, (z)_y) \in R']. \end{aligned}$$

Clearly the right side of (1) implies the left side. If the left side holds, choose for each $y < x_{n-1}$ an integer $w_y \in \omega$ such that $(x_0, \ldots, x_{n-2}, y, w_y) \in R'$, and let $z = \Pi_{y < x(n-1)} \mathrm{p}_y^{wy}$; clearly then z is as desired in the right side of (1). □

The following proposition is obvious:

Proposition 5.32. *If R is an n-ary relation, then $R \in \Sigma_m$ iff ${}^n\omega \sim R \in \Pi_m$.*

Proposition 5.33. $\Sigma_m \cup \Pi_m \subseteq \Delta_{m+1}$.

PROOF. Let $R \in \Sigma_m$, say R is n-ary. Let $S = \{(x_0, \ldots, x_n) : (x_0, \ldots, x_{n-1}) \in R\}$. Then $S \in \Sigma_m$ by 5.26 and 5.27. Clearly $R = \{(x_0, \ldots, x_{n-1}) : \forall y \in \omega[(x_0, \ldots, x_{n-1}, y) \in S]\}$, so $R \in \Pi_{m+1}$. Thus $\Sigma_m \subseteq \Pi_{m+1}$, and similarly $\Pi_m \subseteq \Sigma_{m+1}$. An easy inductive argument shows that $\Sigma_m \subseteq \Sigma_{m+1}$ and $\Pi_m \subseteq \Pi_{m+1}$. □

We will return to the following important result several times later on:

Theorem 5.34. $\Delta_1 = \Delta_0$.

PROOF. We know that $\Delta_0 \subseteq \Delta_1$. Suppose $R \in \Delta_1$, say R is n-ary. Then there are recursive S, $T((n + 1)$-ary) such that for all $x_0, \ldots, x_{n-1} \in \omega$,

$$\begin{aligned}(x_0, \ldots, x_{n-1}) \in R \quad & \text{iff } \exists y[(x_0, \ldots, x_{n-1}, y) \in S] \\ & \text{iff } \forall y[(x_0, \ldots, x_{n-1}, y) \in T].\end{aligned}$$

Hence, as is easily seen,

$$\begin{aligned}&\chi_R(x_0, \ldots, x_{n-1}) \\ &\qquad = \chi_S(x_0, \ldots, x_{n-1}, \mu y[(x_0, \ldots, x_{n-1}, y) \in S \text{ or } (x_0, \ldots, x_{n-1}, y) \notin T]),\end{aligned}$$

so R is recursive. □

Intuitively, to determine whether or not $(x_0, \ldots, x_{n-1}) \in R$ we check in succession $(x_0, \ldots, x_{n-1}, 0)$, $(x_0, \ldots, x_{n-1}, 1), \ldots$ for membership in S and T. Eventually one of these is in S (hence $(x_0, \ldots, x_{n-1}) \in R$), or else one of them fails to be in T (hence $(x_0, \ldots, x_{n-1}) \notin R$).

Now we extend our normal form results up into the arithmetical hierarchy:

Theorem 5.35. *For $m, n > 0$ there is an $(n + 1)$-ary Σ_m-relation R^n_m with the following properties:*

(*i*) *for every n-ary Σ_m-relation S there is an $e \in \omega$ such that $S = \{(x_0, \ldots, x_{n-1}) : (e, x_0, \ldots, x_{n-1}) \in R^n_m\}$;*
(*ii*) *for every n-ary Π_m-relation S there is an $e \in \omega$ such that $S = \{(x_0, \ldots, x_{n-1}) : (e, x_0, \ldots, x_{n-1}) \notin R^n_m\}$.*

PROOF. We construct R^n_m by recursion on m. Let

$$R^n_1 = \{(x_0, \ldots, x_n) : \exists y \in \omega[(x_0, \ldots, x_n, y) \in \mathrm{T}_n]\}.$$

If R^n_m has been defined for all n, let

$$R^n_{m+1} = \{(x_0, \ldots, x_n) : \exists y \in \omega[(x_0, \ldots, x_n, y) \notin R^{n+1}_m]\}.$$

It is easily seen by induction on m, using 5.23, that the desired conditions hold. $\square$

Theorem 5.36 (Hierarchy theorem). *For any $m, n > 0$ there exists an n-ary relation $T \in \Sigma_m \sim \Pi_m$. Hence ${}^n\omega \sim T \in \Pi_m \sim \Sigma_m$. Furthermore, there is an n-ary relation $W \in \Delta_{m+1} \sim (\Sigma_m \cup \Pi_m)$.*

PROOF. Let R^n_m be as in 5.35. Let

$$T = \{(x_0, \ldots, x_{n-1}) : (x_0, x_0, x_1, x_2, \ldots, x_{n-1}) \in R^n_m\}.$$

Thus $T \in \Sigma_m$. If $T \in \Pi_m$, by 5.35 choose $e \in \omega$ so that $T = \{(x_0, \ldots, x_{n-1}) : (e, x_0, \ldots, x_{n-1}) \notin R^n_m\}$. Then

$$e^{(n+1)} \in R^n_m \qquad \text{iff } e^{(n)} \in T \qquad \text{iff } (e)^{n+1} \notin R^n_m,$$

a contradiction. Thus $T \notin \Pi_m$.

For the second part of the theorem, let T be as in the first part. Set

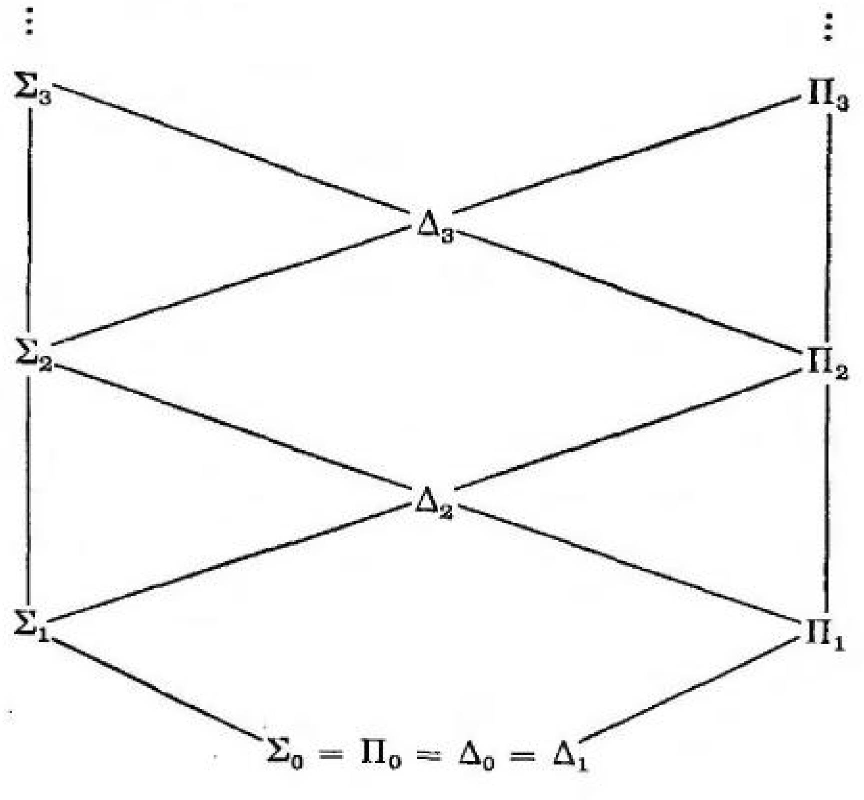
$$W = \{(x_0, \ldots, x_{n-1}) : ((x_0)_0, (x_1)_0, \ldots, (x_{n-1})_0) \notin T \qquad \text{and} \\ ((x_0)_1, (x_1)_1, \ldots, (x_{n-1})_1) \in T\}.$$

Now T, ${}^n\omega \sim T \in \Delta_{m+1}$ by 5.33, so $W \in \Delta_{m+1}$. Suppose $W \in \Sigma_m$. Choose $(t_0, \ldots, t_{n-1}) \in T$ (T is obviously nonempty since $0 \in \Pi_m$). For any $x_0, \ldots, x_{n-1} \in \omega$ we have

$$(x_0, \ldots, x_{n-1}) \notin T \qquad \text{iff } (2^{x0} \cdot 3^{t0}, \ldots, 2^{x(n-1)} \cdot 3^{t(n-1)}) \in W,$$

so ${}^n\omega \sim T \in \Sigma_m$, contradiction. Similarly, $W \in \Pi_m$ leads to a contradiction. $\square$

Thus the arithmetical hierarchy appears as in the following diagram, where the lines indicate proper inclusions:



BIBLIOGRAPHY

1. Kleene, S. C. *Introduction to Metamathematics*. Princeton: van Nostrand (1952).
2. Malcev, A. I. *Algorithms and Recursive Functions*. Groningen: Wolters-Noordhoff (1970).
3. Rogers, H. *Theory of Recursive Functions and Effective Computability*. New York: McGraw-Hill (1967).

EXERCISES

5.37. If f is a finite function (i.e., it is a finite set and is a function), then f is partial recursive.

5.38. Give an example of a binary partial recursive function f such that if g is defined by

$$\begin{aligned} gx &= \text{least } y \text{ such that } f(x, y) \text{ is defined and } f(x, y) = 0, \\ gx &= \text{undefined if no such } y, \end{aligned}$$

then g is *not* partial recursive (cf. 5.3 and following remarks).

5.39. Give an example of a binary recursive function f such that if g is defined by

$$\begin{aligned} gx &= \text{least } y \text{ such that } f(x, y) = 0, \\ gx &= 0 \qquad \text{if no such } y, \end{aligned}$$

then g is *not* recursive.

5.40′. The class of partial recursive functions is the intersection of all classes C of partial functions such that $\mathfrak{s} \in C$, $U_i^n \in C$ whenever $i < n$, and C is closed under composition, primitive recursion, and minimalization, all except composition applied only to total functions.

5.41. If f is an m-ary partial recursive function and Dmn f is recursive, then f can be extended to a general recursive function.

5.42. Give an example of an m-ary partial recursive function f which can be extended to a general recursive function, but has the property that Dmn f is not recursive.

5.43. There is a unary partial recursive function f such that for no binary recursive function g is it true that for all x, $fx \simeq \mu y[g(x, y) = 0]$. *Hint:* let $fx \simeq \varphi_x^1 x \cdot 0 + x$ for all x. If g works as above, let $hx = \varphi_x^1 x + 1$ if $g(x, x) = 0$, $hx = 0$ otherwise. Show h is recursive and obtain a contradiction.

5.44. For any total function f of one variable the following conditions are equivalent:

(1) there is a recursive function g of two variables such that for all $x \in \omega$, $fx = \mu y[g(x, y) = 0]$.

(2) $\{(x, fx) : x \in \omega\}$ is a recursive relation.

The conditions remain equivalent if in both (1) and (2) "recursive" is replaced by "primitive recursive" or by "elementary."

5.45. If f is a unary recursive function, then $\{(x, fx) : x \in \omega\}$ is a recursive relation. Similarly if we replace both words "recursive" by "primitive recursive" or by "elementary."

5.46. Give an example of a unary partial recursive function f such that $\{(x, fx) : x \in \mathrm{Dmn}\, f\}$ is not recursive.

5.47. There is a recursive set which is not elementary.

5.48. There is a unary recursive function f for which there is no binary elementary function g such that for all $x \in \omega$, $fx = \mu y[g(x, y) = 0]$. *Hint:* take $f = \chi_A$, where A is as in 5.47.

5.49. There is a total unary function f such that $\{(x, fx) : x \in \omega\}$ is elementary but f is not elementary.

5.50. There is no recursive procedure for deciding for an arbitrary e whether or not φ_e^1 has infinite range.

5.51. Assume $m > 1$. Let $A = \{e : \varphi_e^m$ is a special recursive function$\}$. Show that A is not recursive.

5.52. Show that the function f defined as follows is recursive.

$$\begin{aligned} f(0, y) &= y + 1, \\ f(1, y) &= y + 2, \\ f(x + 2, 0) &= f(x + 1, 1), \\ f(x + 2, y + 1) &= f(x, f(x + 1, f(x + 2, y))). \end{aligned}$$

5.53. Show that there is no recursive function f satisfying the following conditions:

$$\begin{aligned} f(0, y) &= y + 2, \\ f(x + 1, 0) &= f(x, 1), \\ f(x + 1, y + 1) &= f(x + 1, f(x, y)) + 1. \end{aligned}$$

6 Recursively Enumerable Sets

In this chapter we shall deal in some detail with the set Σ_1 of relations (see 5.24). Such relations are called *recursively enumerable* for reasons which will shortly become clear. The study of recursively enumerable relations is one of the main branches of recursive function theory. They play a large role in logic. In fact, for most theories the set of Gödel numbers of theorems is recursively enumerable. Thus many of the concepts introduced in this section will have applications in our discussion of decidable and undecidable theories in Part III. Unless otherwise stated, the functions in this chapter are unary.

A nonempty set is *effectively enumerable* provided there is an automatic method for listing out its members, one after the other. This does *not* imply that there is a decision method for determining membership in the set. The formal version of this notion is given in

Definition 6.1. A set $A \subseteq \omega$ is *recursively enumerable* (for brevity *r.e.*) if $A = 0$ or A is the range of a recursive function.

This definition can be given several equivalent forms, each having its own intuitive appeal:

Theorem 6.2. *For $A \subseteq \omega$ the following are equivalent;*

(*i*) *$A = 0$ or A is the range of an elementary function;*
(*ii*) *$A = 0$ or A is the range of a primitive recursive function;*
(*iii*) *A is recursively enumerable;*
(*iv*) *A is the range of a partial recursive function;*
(*v*) *A is the domain of a partial recursive function;*
(*vi*) *$A \in \Sigma_1$.*

PROOF. Obviously $(i) \Rightarrow (ii) \Rightarrow (iii)$. To show that $(iii) \Rightarrow (iv)$ we just need to show that 0 (the empty set) is the range of some partial recursive function; and obviously the only possibility for such a function is 0 (which is also the empty function). 0 is partial recursive by the argument following 5.3.

$(iv) \Rightarrow (v)$. Let $A = \text{Rng } \varphi_e^1$. For any $x \in \omega$ let

$$fx \simeq \mu y((e, (y)_0, (y)_1) \in \mathrm{T}_1 \text{ and } \mathrm{V}(y)_1 = x).$$

Clearly then $\text{Dmn } f = \text{Rng } \varphi_e^1 = A$, and f is partial recursive.

$(v) \Rightarrow (vi)$. Suppose $A = \text{Dmn } \varphi_e^1$. Then for all $x \in \omega$, $x \in A$ iff $\exists y((e, x, y) \in \mathrm{T}_1)$, so $A \in \Sigma_1$.

$(vi) \Rightarrow (i)$. Suppose $A \in \Sigma_1$. By 5.23 choose $e \in \omega$ such that $A = \{x : \exists y((e, x, y) \in \mathrm{T}_1)\}$. We may assume that $A \neq 0$; say $a \in A$. Now for any $x \in \omega$ let

$$\begin{aligned} fx &= (x)_0 \quad && \text{if } (e, (x)_0, (x)_1) \in \mathrm{T}_1, \\ fx &= a && \text{otherwise.} \end{aligned}$$

Clearly f is an elementary function and $\text{Rng } f = A$, as desired. □

An intuitive proof of the equivalence of 6.2(*iii*) and 6.2(*v*) is instructive. First assume that A is recursively enumerable, $A \neq 0$. Say $A = \text{Rng } f$, recursive. We define a function g with domain A as follows. To calculate gx, we look along the list $f0, f1, \ldots$ for x. If we find it, we set $gx = 0$. If x is never found, gx is never computed. Clearly g is effectively calculable (see introduction to Chapter 5), and $\text{Dmn } g = A$.

Conversely, suppose $A = \text{Dmn } g$, g partial recursive, and assume that $A \neq 0$. Now we make the following calculations:

two steps in the calculation of $g0$
one step in the calculation of $g1$
three steps in the calculation of $g0$
two steps in the calculation of $g1$
one step in the calculation of $g2$
four steps in the calculation of $g0$
three steps in the calculation of $g1$
two steps in the calculation of $g2$
one step in the calculation of $g3$
$\vdots$

During this process we will occasionally obtain answers. At regular intervals we list out all the x for which we have so far calculated gx. Since $A \neq 0$, eventually we will list at least one x, and then at regular intervals we put more on our list (with many repetitions). Calling the list $f0, f1, \ldots$, clearly f is an effectively calculable total function with range A.

Now we want to investigate the relationship between recursive and recursively enumerable sets. By 5.33 and 5.36 we have

Theorem 6.3. *Every recursive set is recursively enumerable. There is a recursively enumerable set which is not recursive.*

The second part of 6.3 is one of the most important results of recursion theory, so we give its proof here in a more direct form:

Definition 6.4. $K = \{x : \exists y((x, x, y) \in \mathrm{T}_1)\}$.

Theorem 6.5. *K is recursively enumerable but not recursive.*

PROOF. Obviously $\mathrm{K} \in \Sigma_1$ so K is recursively enumerable. Suppose K is recursive. Then so is $\omega \sim \mathrm{K}$, so by 6.2(v) there is an $e \in \omega$ such that $\omega \sim \mathrm{K} = \mathrm{Dmn}\ \boldsymbol{\varphi}_e^1$. Then

$$\begin{array}{ll} e \in \mathrm{K} & \text{iff } e \in \mathrm{Dmn}\ \boldsymbol{\varphi}_e^1 \text{ by the definition of K,} \\ e \notin \mathrm{K} & \text{iff } e \in \mathrm{Dmn}\ \boldsymbol{\varphi}_e^1 \text{ by the choice of } e, \end{array}$$

contradiction. □

The set K will be discussed further later on. Another important relationship between recursive and recursively enumerable sets is given in 5.34, which can be reformulated as follows:

Theorem 6.6. *Let $A \subseteq \omega$. The following conditions are equivalent:*

(i) *A is recursive;*
(ii) *A and $\omega \sim A$ are recursively enumerable.*

This theorem can be seen in the following fashion, working directly from Definition 6.1: Of course (ii) $\Rightarrow$ (i) is the main part of 6.6. Assume (ii). We may suppose $0 \neq A \neq \omega$. Then let f and g be recursive functions with $\mathrm{Rng}\, f = A$, $\mathrm{Rng}\, g = \omega \sim A$. To determine whether $x \in A$ or not, list out $f0, g0, f1, g1, \ldots$. Eventually x will appear in the list; if $x = fn$ for some n, then $x \in A$, while if $x = gn$ for some n, then $x \in A$. Formally, for any $x \in \omega$,

$$\begin{array}{ll} \chi_A x = 1 & \text{if } f\mu y(fy = x \text{ or } gy = x) = x, \\ \chi_A x = 0 & \text{otherwise.} \end{array}$$

Theorem 6.7. *Let $A \subseteq \omega$. The following are equivalent:*

(i) *A is infinite and recursive:*
(ii) *there is a recursive function f with $\mathrm{Rng}\, f = A$ and $\forall x \in \omega(fx < f(x + 1))$.*

PROOF. (i) $\Rightarrow$ (ii). Let a be the least member of A. Define

$$\begin{aligned} f0 &= a \\ f(x + 1) &= \mu y(y \in A \text{ and } y > fx). \end{aligned}$$

Clearly f is as desired.

(ii) $\Rightarrow$ (i). Assume f as in (ii). Then by induction on x,

(1) $$\forall x \in \omega \qquad (x \leq fx).$$

Thus

(2) $$\forall y \in \mathrm{Rng}\, f\, \exists x \leq y \qquad (fx = y).$$

Hence for all $y \in \omega$,

$$\begin{aligned} \chi_A y &= 1 \qquad \text{if } \exists x \leq y(fx = y) \\ \chi_A y &= 0 \qquad \text{otherwise,} \end{aligned}$$

as desired. □

Theorem 6.8. *Any infinite recursively enumerable set has an infinite recursive subset.*

PROOF. Let A be infinite r.e., say $A = \operatorname{Rng} f$, f recursive. We define g by induction:

$$\begin{aligned} g0 &= f0 \\ g(x + 1) &= f\mu y(fy > gx). \end{aligned}$$

Thus $gx < g(x + 1)$ for all $x \in \omega$, and hence, by 6.7, $\operatorname{Rng} g$ is infinite and recursive. Obviously $\operatorname{Rng} g \subseteq A$. □

Next, we want to investigate closure properties of the class of r.e. sets. Which operations on sets lead out of the class, and under which operation is the class closed? By 5.29, the class of r.e. sets is closed under union and intersection. We can give intuitive proofs of these facts directly from the definition. Let A and B be r.e. sets, and ignore the case when one of them is empty. Let f and g be recursive functions enumerating A and B respectively. One enumerates $A \cup B$ by: $f0, g0, f1, g1, \ldots$. One can enumerate $A \cap B$ by looking along this list and putting a number on a separate list as soon as it appears at both an odd and even step. Both of these procedures can be given a rigorous formulation.

By 6.3 and 6.6, the class of r.e. sets is not closed under complementation. Some further closure properties:

Theorem 6.9. *If A is r.e. and f is partial recursive, then f^*A is r.e.*

PROOF. We may assume that $A \neq 0$. Say $A = \operatorname{Rng} g$, g recursive. Clearly $f^*A = \operatorname{Rng} (f \circ g)$ and $f \circ g$ is partial recursive. □

Theorem 6.10. *If A is r.e. and f is partial recursive, then $f^{-1*}A$ is r.e.*

PROOF. Say $A = \operatorname{Dmn} \varphi_e^1$. Then $f^{-1*}A = \operatorname{Dmn} (\varphi_e^1 \circ f)$ as desired. □

Theorem 6.11. *If A is r.e., then $\bigcup_{x \in A} \operatorname{Rng} \varphi_e^1$ is r.e.*

PROOF. For any $y \in \omega$,

$$y \in \bigcup_{x \in A} \operatorname{Rng} \varphi_x^1 \qquad \text{iff } \exists x \in A(y \in \operatorname{Rng} \varphi_x^1).$$

Since both A and each $\operatorname{Rng} \varphi_x^1$ are in Σ_1, it follows easily that $\bigcup_{x \in A} \operatorname{Rng} \varphi_x^1$ is in Σ_1. □

Before carrying the theory of r.e. sets further we wish to back up and extend our results obtained so far to relations.

Definition 6.12. A relation $R \subseteq {}^m\omega$ is *recursively enumerable* (for brevity *r.e.*) if $A = 0$ or there exist m recursive functions $f_0, \ldots, f_{m-1}$ such that

$$R = \{(f_0x, \ldots, f_{m-1}x) : x \in \omega\}.$$

Theorem 6.13. *For $R \subseteq {}^m\omega$ the following are equivalent:*

(*i*) *$R = 0$ or there exist elementary functions $f_0, \ldots, f_{m-1}$ with $R = \{(f_0x, \ldots, f_{m-1}x) : x \in \omega\}$;*
(*ii*) *like (i) with "elementary" replaced by "primitive recursive";*
(*iii*) *R is recursively enumerable;*
(*iv*) *there exist partial recursive functions $f_0, \ldots, f_{m-1}$ with $R = \{(f_0x, \ldots, f_{m-1}x) : x \in \mathrm{Dmn}\, f_0 \cap \cdots \cap \mathrm{Dmn}\, f_{m-1}\}$;*
(*v*) *$R = 0$ or there is an elementary function f with $R = \{((fx)_0, \ldots, (fx)_{m-1}) : x \in \omega\}$;*
(*vi*) *like (v), with "elementary" replaced by "primitive recursive";*
(*vii*) *like (v), with "elementary" replaced by "recursive";*
(*viii*) *there is a partial recursive function f such that $R = \{((fx)_0, \ldots, (fx)_{m-1}) : x \in \mathrm{Dmn}\, f\}$;*
(*ix*) *there is an m-ary partial recursive function f such that $R = \mathrm{Dmn}\, f$;*
(*x*) *$R \in \Sigma_1$.*

PROOF. Clearly (*i*) $\Rightarrow$ (*ii*) $\Rightarrow$ (*iii*) $\Rightarrow$ (*iv*).

(*iv*) $\Rightarrow$ (*v*). Assume (*iv*), with $f_0, \ldots, f_{m-1}$ partial recursive and $R = \{(f_0x, \ldots, f_{m-1}x) : x \in \mathrm{Dmn}\, f_0 \cap \cdots \cap \mathrm{Dmn}\, f_{m-1}\}$. We may assume that $R \neq 0$, say $(a_0, \ldots, a_{m-1}) \in R$. Say $f_0 = \varphi^1_{e0}, \ldots, f_{m-1} = \varphi^1_{e(m-1)}$. For any $x \in \omega$, let

$$\begin{aligned} gx &= \prod_{i<m} \mathrm{p}_i^{V((x)(i+1))} && \text{if } (e_i, (x)_0, (x)_{i+1}) \in \mathrm{T}_1 \text{ for all } i < m, \\ gx &= \prod_{i<m} \mathrm{p}_i^{ai} && \text{otherwise.} \end{aligned}$$

Clearly g is elementary and $R = \{((gx)_0, \ldots, (gx)_{m-1}) : x \in \omega\}$.

Obviously (*v*) $\Rightarrow$ (*vi*) $\Rightarrow$ (*vii*) $\Rightarrow$ (*viii*).

(*viii*) $\Rightarrow$ (*ix*). Suppose f is as in (*viii*). Say $f = \varphi^1_e$. For any $x_0, \ldots, x_{m-1} \in \omega$ let

$$g(x_0, \ldots, x_{m-1}) = \mu y((e, (y)_0, (y)_1) \in \mathrm{T}_1 \text{ and } \mathrm{V}(y)_1 = \prod_{i<m} \mathrm{p}_i^{xi}).$$

Clearly g is partial recursive and $R = \mathrm{Dmn}\, g$.

(*ix*) $\Rightarrow$ (*x*). Suppose $R = \mathrm{Dmn}\, \varphi^m_e$. Then

$$R = \{(x_0, \ldots, x_{m-1}) : \exists y((e, x_0, \ldots, x_{m-1}, y) \in \mathrm{T}_m)\},$$

so $R \in \Sigma_1$.

(*x*) $\Rightarrow$ (*i*). Suppose $R \in \Sigma_1$. By 5.23, choose $e \in \omega$ such that

$$R = \{(x_0, \ldots, x_{m-1}) : \exists y((e, x_0, \ldots, x_{m-1}, y) \in \mathrm{T}_m)\}.$$

We may assume that $R \neq 0$; say $(a_0, \ldots, a_{m-1}) \in R$. Now for $i < m$ and any $x \in \omega$ let

$$\begin{aligned} f_i x &= (x)_i && \text{if } (e, (x)_0, \ldots, (x)_m) \in \mathrm{T}_m \\ f_i x &= a_i && \text{otherwise.} \end{aligned}$$

Clearly each f_i is elementary and $R = \{(f_0 x, \ldots, f_{m-1} x) : x \in \omega\}$. □

Theorem 6.14. *Every recursive relation is recursively enumerable. For each positive m there is a recursively enumerable m-ary relation which is not recursive.*

PROOF. The first part is true by 6.13(x) and 5.33; for the second part, use 5.36. □

The following result is proved just as for sets.

Theorem 6.15. *Let $R \subseteq {}^m\omega$. The following conditions are equivalent:*

(*i*) *R is recursive;*
(*ii*) *R and ${}^m\omega \sim R$ are recursively enumerable.*

The following important theorem shows that the notion of a partial recursive function can be defined without resorting to the rather complicated notions discussed at the beginning of Chapter 5.

Theorem 6.16. *Let f be a unary partial function. Then the following conditions are equivalent:*

(*i*) *f is partial recursive;*
(*ii*) *$\{(x, fx) : x \in \mathrm{Dmn}\, f\}$ is r.e.*

PROOF. (i) $\Rightarrow$ (ii). Assume (i). For any $x, y \in \omega$ let

$$g(x, y) \simeq \mu z(|y - fx| = 0).$$

Clearly g is partial recursive and $\mathrm{Dmn}\, g = \{(x, fx) : x \in \mathrm{Dmn}\, f\}$.

(ii) $\Rightarrow$ (i). Assume (ii), and by 6.12 let g and h be recursive functions such that

$$\{(x, fx) : x \in \mathrm{Dmn}\, f\} = \{(gx, hx) : x \in \omega\}.$$

Then for any $x \in \omega$,

$$fx \simeq h\mu y(gy = x),$$

so f is partial recursive. □

We now turn to the study of some special r.e. sets.

Definition 6.17

(*i*) A set $A \subseteq \omega$ is *productive* if there is a recursive function f (called a *productive function for A*) such that for all $e \in \omega$, if $\mathrm{Dmn}\, \varphi_e^1 \subseteq A$ then $fe \in A \sim \mathrm{Dmn}\, \varphi_e^1$.

(*ii*) A set $A \subseteq \omega$ is *creative* if A is r.e. and $\omega \sim A$ is productive.

Thus a productive set A is strongly not recursively enumerable: there is an effective procedure for finding members of $A \sim B$ for any r.e. subset B of A. A creative set, while r.e., is strongly nonrecursive. The sets of Gödel numbers of theorems of many theories studied in Part III are creative, as we shall see.

Recall Definition 6.4.

Theorem 6.18. K *is creative.*

PROOF. By 6.5, K is r.e. Now U_0^1 is a productive function for $\omega \sim \mathrm{K}$. For if $e \in \omega$ and $\mathrm{Dmn}\, \boldsymbol{\varphi}_e^1 \subseteq \omega \sim \mathrm{K}$, then $e \in (\omega \sim \mathrm{K}) \sim \mathrm{Dmn}\, \boldsymbol{\varphi}_e^1$; for

$$\begin{aligned} e \in \mathrm{K} &\Rightarrow e \in \mathrm{Dmn}\, \boldsymbol{\varphi}_e^1 && \text{by definition of K, 6.4} \\ &\Rightarrow e \in \omega \sim \mathrm{K} && \text{by assumption } \mathrm{Dmn}\, \boldsymbol{\varphi}_e^1 \subseteq \omega \sim \mathrm{K} \end{aligned}$$

so $e \in \omega \sim \mathrm{K}$, and hence by definition of K, $e \notin \mathrm{Dmn}\, \boldsymbol{\varphi}_e^1$. □

The next theorem shows that, in a sense, any r.e. set can be obtained from a creative set; cf. 6.10 and the initial section of Chapter 7.

Theorem 6.19. *If A is r.e. and C is creative, then there is a recursive function f such that $A = f^{-1*}C$.*

PROOF. Say $A = \mathrm{Dmn}\, \boldsymbol{\varphi}_d^1$, and let g be a productive function for $\omega \sim C$. For any $x, y, z \in \omega$ let

$$l(z, y, x) \simeq \mu u[z = g\mathrm{s}_1^1(x, y)] + \boldsymbol{\varphi}_d^1 y.$$

Thus l is partial recursive. By the recursion theorem (5.15) choose $e \in \omega$ such that for all $y, z \in \omega$,

$$l(z, y, e) \simeq \boldsymbol{\varphi}_e^2(z, y).$$

Let $fy = g\mathrm{s}_1^1(e, y)$ for all $y \in \omega$. We claim that $A = f^{-1*}C$. Since f is obviously recursive, this will complete the proof.

First suppose that $y \in A$. Then

$$\text{(1)} \qquad \mathrm{Dmn}\, \boldsymbol{\varphi}^1(\mathrm{s}_1^1(e, y)) = \{g\mathrm{s}_1^1(e, y)\}.$$

In fact, by 5.13 we have

$$\begin{aligned} z \in \mathrm{Dmn}\, \boldsymbol{\varphi}^1(\mathrm{s}_1^1(e, y)) \quad &\text{iff } (z, y) \in \mathrm{Dmn}\, \boldsymbol{\varphi}_e^2 \\ &\text{iff } (z, y, e) \in \mathrm{Dmn}\, l \text{ (by choice of } e) \\ &\text{iff } z = g\mathrm{s}_1^1(e, y) \text{ and } y \in \mathrm{Dmn}\, \boldsymbol{\varphi}_d^1 \\ &\text{iff } z = g\mathrm{s}_1^1(e, y). \end{aligned}$$

Thus (1) holds. Now if $fy \notin C$, this means that $g\mathrm{s}_1^1(e, y) \notin C$ and so by (1) $\mathrm{Dmn}\, \boldsymbol{\varphi}^1(\mathrm{s}_1^1(e, y)) \subseteq \omega \sim C$. Since g is a productive function for $\omega \sim C$ we would get

$$g\mathrm{s}_1^1(e, y) \in (\omega \sim C) \sim \mathrm{Dmn}\, \boldsymbol{\varphi}^1(\mathrm{s}_1^1(e, y)),$$

contradicting (1). Thus $fy \in C$.

Second, suppose that $y \notin A$. Then $y \notin \mathrm{Dmn}\, \varphi_a^1$, so $\forall z\, (z, y, e) \notin \mathrm{Dmn}\, l$, hence $\forall z((z, y) \notin \mathrm{Dmn}\, \varphi_e^2)$, so by 5.13 $\mathrm{Dmn}\, \varphi^1(\mathrm{s}_1^1(e, y)) = 0$. Thus, since g is productive,

$$fy = g\mathrm{s}_1^1(e, y) \in (\omega \sim C) \sim \mathrm{Dmn}\, \varphi^1(\mathrm{s}_1^1(e, y)),$$

in particular $fy \notin C$, as desired. □

The following result will not play a role in our logical discussion, but is important in the general theory of r.e. sets. See also the definition and results concerning simple sets below.

Theorem 6.20. *If A is productive, then A has an infinite recursive subset.*

PROOF. By 6.8 it suffices to show that A has an infinite r.e. subset. Let f be a productive function for A. For any x, y, let

$$k(y, x) \simeq \mu i(i \leq \mathrm{l}x \text{ and } y = (x)_i \dot{-} 1).$$

Clearly k is partial recursive; say $k = \varphi_e^2$. Now for any $x \in \omega$,

(1) $$\mathrm{Dmn}\, \varphi^1(\mathrm{s}_1^1(e, x)) = \{(x)_i \dot{-} 1 : i \leq \mathrm{l}x\}.$$

In fact, for any $y \in \omega$,

$$\begin{aligned} y \in \mathrm{Dmn}\, \varphi^1(\mathrm{s}_1^1(e, x)) \quad &\text{iff } (y, x) \in \mathrm{Dmn}\, \varphi_e^2 \quad \text{iff } (y, x) \in \mathrm{Dmn}\, k \\ &\text{iff } \exists i \leq \mathrm{l}x(y = (x)_i \dot{-} 1). \end{aligned}$$

Now let r be such that $\varphi_r^1 = 0$, and define

$$\begin{aligned} g(x, y) &= fr \quad \text{if } y = 0 \text{ or } 1, \\ g(x, y) &= f\mathrm{s}_1^1(e, y) \quad \text{if } y \neq 0 \text{ and } y \neq 1. \end{aligned}$$

Thus g is recursive. Now define $t\colon \omega \to \omega$ by setting, for any $x \in \omega$, $tx = g(x, \bar{t}x)$. Here $\bar{t}x$ is defined in 2.31, and by 2.33, t is recursive. Now we claim for all $x \in \omega$,

(2) $$tx \in A \sim \{ty : y < x\}.$$

We establish (2) by induction on x. For $x = 0$,

$$t0 = g(0, \bar{t}0) = g(0, 1) = fr \in A$$

(since $\varphi_r^1 = 0 \subseteq A$ and f is a productive function for A). Thus (2) holds for $x = 0$. Suppose (2) holds for all $x' < x$, where $x \neq 0$. Then $tx = g(x, \bar{t}x)$, and $\bar{t}x \neq 0, 1$, so $tx = f\mathrm{s}_1^1(e, \bar{t}x)$. Also

$$\mathrm{Dmn}\, \varphi^1(\mathrm{s}_1^1(e, \bar{t}x)) = \{ty : y < x\} \subseteq A$$

by (1) and the induction hypothesis. Since f is a productive function for A, $tx = f\mathrm{s}_1^1(e, \bar{t}x) \in A \sim \{ty : y < x\}$, as desired. Thus (2) holds. Hence Rng t is an infinite r.e. subset of A, and the proof is complete. □

We now give a method to arrive at creative sets.

Definition 6.21

(*i*) Two sets A and B are *recursively separable* if there is a recursive set C such that $A \subseteq C$ and $B \subseteq \omega \sim C$.

(*ii*) A and B are *recursively inseparable* if they are disjoint but not recursively separable.

(*iii*) A and B are *effectively inseparable* if they are disjoint and there is a 2-ary recursive function f such that for all e and r, if $A \subseteq \text{Dmn}\,\varphi^1_e$, $B \subseteq \text{Dmn}\,\varphi^1_r$, and $\text{Dmn}\,\varphi^1_e \cap \text{Dmn}\,\varphi^1_r = 0$, then $f(e, r) \in \omega \sim (\text{Dmn}\,\varphi^1_e \cup \text{Dmn}\,\varphi^1_r)$.

Effectively inseparable sets will be constructed in abundance in Part III; most undecidability results actually yield such sets.

Obviously we have:

Theorem 6.22. *If A and B are effectively inseparable then they are recursively inseparable.*

The converse of 6.22 fails; see Exercises 6.47, 6.48.

Theorem 6.23. *If A and B are recursively enumerable and effectively inseparable, then both A and B are creative.*

Proof. By symmetry it suffices to show that A is creative, i.e. that $\omega \sim A$ is productive. Let f be as in 6.21(*iii*). Say $A = \text{Dmn}\,\varphi^1_u$ and $B = \text{Dmn}\,\varphi^1_s$. For any $e, x \in \omega$ let

$$g(x, e) \simeq \mu y((e, x, y) \in \mathrm{T}_1 \text{ or } (s, x, y) \in T_1).$$

Thus $\text{Dmn}\,g = \{(x, e) : x \in \text{Dmn}\,\varphi^1_e \cup B\}$. Clearly g is partial recursive; say $g = \varphi^2_r$. Now for any $e \in \omega$ we have, by 5.13,

$$(1) \qquad \text{Dmn}\,\varphi^1(\mathrm{s}^1_1(r, e)) = \{x : (x, e) \in \text{Dmn}\,\varphi^2_r\} = \text{Dmn}\,\varphi^1_e \cup B.$$

Let for any $e \in \omega$ $he = f(u, \mathrm{s}^1_1(r, e))$. Thus h is recursive; we claim it is a productive function for $\omega \sim A$. In fact, suppose $\text{Dmn}\,\varphi^1_e \subseteq \omega \sim A$. Then, using (1), $A = \text{Dmn}\,\varphi^1_u$, $B \subseteq \text{Dmn}\,\varphi^1(\mathrm{s}^1_1(r, e))$, and $\text{Dmn}\,\varphi^1_u \cap \text{Dmn}\,\varphi^1(\mathrm{s}^1_1(r, e)) = 0$. Hence, by 6.21(*iii*), $he = f(u, \mathrm{s}^1_1(r, e)) \in \omega \sim (\text{Dmn}\,\varphi^1_u \cup \text{Dmn}\,\varphi^1(\mathrm{s}^1_1(r, e)))$, i.e., $he \in \omega \sim (A \cup \text{Dmn}\,\varphi^1_e \cup B)$, so $he \in (\omega \sim A) \sim \text{Dmn}\,\varphi^1_e$, as desired. □

Theorem 6.24. *There exist two recursively enumerable effectively inseparable sets.*

Proof. Let

$$K_1 = \{x : \exists y[((x)_0, x, y)\ \mathrm{T}_1 \text{ and } \forall z \leq y(((x)_1, x, z) \notin \mathrm{T}_1)]\},$$
$$K_2 = \{x : \exists y[((x)_1, x, y) \in \mathrm{T}_1 \text{ and } \forall z \leq y(((x)_0, x, z) \notin \mathrm{T}_1)]\}.$$

Clearly K_1 and K_2 are r.e. and $K_1 \cap K_2 = 0$. For any $e, r \in \omega$ let $f(e, r) = 2^r \cdot 3^e$. To verify 6.21(*iii*), assume that $K_1 \subseteq \mathrm{Dmn}\, \varphi_e^1$ and $K_2 \subseteq \mathrm{Dmn}\, \varphi_r^1$ with $\mathrm{Dmn}\, \varphi_e^1 \cap \mathrm{Dmn}\, \varphi_r^1 = 0$. Suppose $f(e, r) \in \mathrm{Dmn}\, \varphi_e^1 \cup \mathrm{Dmn}\, \varphi_r^1$. By symmetry, say $f(e, r) \in \mathrm{Dmn}\, \varphi_e^1$. Thus $\exists y((e, 2^r \cdot 3^e, y) \in \mathrm{T}_1)$, and since $\mathrm{Dmn}\, \varphi_e^1 \cap \mathrm{Dmn}\, \varphi_r^1 = 0$, obviously $\forall z((r, 2^r \cdot 3^e, z) \notin \mathrm{T}_1)$. Thus $2^r \cdot 3^e \in K_2$, so $2^r \cdot 3^e \in \mathrm{Dmn}\, \varphi_r^1$, contradiction. □

The next theorem gives an important method of producing new effectively inseparable sets from old ones:

Theorem 6.25. *Suppose that A and B are effectively inseparable, f is a unary recursive function, $C, D \subseteq \omega$, $C \cap D = 0$, $A \subseteq f^{-1*}C$, and $B \subseteq f^{-1*}D$. Then C and D are effectively inseparable.*

Proof. Let h be a function given by 6.21(*iii*) because A and B are effectively inseparable. For any $e, x \in \omega$, let $g(x, e) \simeq \mu y((e, fx, y) \in \mathrm{T}_1)$. Thus g is partial recursive; say $g = \varphi_r^2$. Now we can define a function k intended to satisfy 6.21(*iii*) for C and D: for any $e, u \in \omega$, let $k(e, u) = fh(\mathrm{s}_1^1(r, e), \mathrm{s}_1^1(r, u))$. Thus k is recursive. In order to verify 6.21(*iii*), assume that $C \subseteq \mathrm{Dmn}\, \varphi_e^1$ and $D \subseteq \mathrm{Dmn}\, \varphi_u^1$, where $\mathrm{Dmn}\, \varphi_e^1 \cap \mathrm{Dmn}\, \varphi_u^1 = 0$. It follows that $A \subseteq f^{-1*}\, \mathrm{Dmn}\, \varphi_e^1$, $B \subseteq f^{-1*}\, \mathrm{Dmn}\, \varphi_u^1$, and $f^{-1*}\, \mathrm{Dmn}\, \varphi_e^1 \cap f^{-1*}\, \mathrm{Dmn}\, \varphi_u^1 = 0$. Now for any $x \in \omega$,

$$\begin{aligned} x \in f^{-1*}\, \mathrm{Dmn}\, \varphi_e^1 \quad & \text{iff } fx \in \mathrm{Dmn}\, \varphi_e^1 \\ & \text{iff } \exists y((e, fx, y) \in \mathrm{T}_1) \\ & \text{iff } (x, e) \in \mathrm{Dmn}\, g = \mathrm{Dmn}\, \varphi_r^2 \\ & \text{iff } x \in \mathrm{Dmn}\, \varphi^1 \mathrm{s}_1^1(r, e). \end{aligned}$$

Similarly, $f^{-1*}\, \mathrm{Dmn}\, \varphi_u^1 = \mathrm{Dmn}\, \varphi^1 \mathrm{s}_1^1(r, u)$. Thus $A \subseteq \mathrm{Dmn}\, \varphi^1(\mathrm{s}_1^1(r, e)$, $B \subseteq \mathrm{Dmn}\, \varphi^1 \mathrm{s}_1^1(r, u)$, and $\mathrm{Dmn}\, \varphi^1 \mathrm{s}_1^1(r, e) \cap \mathrm{Dmn}\, \varphi^1 \mathrm{s}_1^1(r, u) = 0$. Hence by choice of h, $h(\mathrm{s}_1^1(r, e), \mathrm{s}_1^1(r, u)) \in \omega \sim (\mathrm{Dmn}\, \varphi^1 \mathrm{s}_1^1(r, e) \cup \mathrm{Dmn}\, \varphi^1 \mathrm{s}_1^1(r, u))$, and hence $k(e, u) \in \omega \sim (\mathrm{Dmn}\, \varphi_e^1 \cup \mathrm{Dmn}\, \varphi_u^1)$, as desired. □

As our final topic in this chapter we briefly consider a kind of r.e. set much different from creative sets. We introduce them partly to give a class of sets which are not creative, and partly because there is a big literature concerning them.

Definition 6.26. A set $A \subseteq \omega$ is *simple* if A is r.e., $\omega \sim A$ is infinite, and $B \cap A \neq 0$ whenever B is an infinite r.e. set.

Theorem 6.27. *A simple set is neither recursive nor creative.*

Proof. If A is simple and recursive, then $\omega \sim A$ is an infinite r.e. set and $A \cap (\omega \sim A) = 0$, contradiction. If A is simple and creative, by 6.20 choose B infinite recursive such that $B \subseteq \omega \sim A$. Contradiction. □

Theorem 6.28. *Simple sets exist.*

PROOF. Let g be a recursive function universal for unary primitive recursive functions (see Lemma 3.5). For any $e \in \omega$ let

$$fe \simeq (\mu y[g(e, (y)_0) = (y)_1 \text{ and } (y)_1 > 2e])_1.$$

Thus f is partial recursive. For each $e \in \omega$ let $\psi_e x = g(e, x)$ for all $x \in \omega$. Clearly for any $e \in \omega$,

(1) if $e \in \operatorname{Dmn} f$, then $fe \in \operatorname{Rng} \psi_e$ and $fe > 2e$;
(2) if $\operatorname{Rng} \psi_e$ is infinite then $e \in \operatorname{Dmn} f$.

Now $\operatorname{Rng} f$ is simple. For, it is obviously r.e. Suppose B is any infinite r.e. set. By choice of g, choose $e \in \omega$ so that $\operatorname{Rng} \psi_e = B$. By (2) and (1), $fe \in \operatorname{Rng} \psi_e$. Thus $B \cap \operatorname{Rng} f \neq 0$. Finally, to show that $\omega \sim \operatorname{Rng} f$ is infinite, note

(3) if $n \in \omega$, then $2n \cap \operatorname{Rng} f \subseteq f^*n$.

For, let $i \in 2n \cap \operatorname{Rng} f$. Say $i = fj$. By (1), $2j < fj$, so $2j < i < 2n$. Thus $j < n$, so $i \in f^*n$.

Since (3) holds, $|2n \cap \operatorname{Rng} f| \leq n$, hence $|2n \sim \operatorname{Rng} f| \geq n$, for any $n \in \omega$. Thus $\omega \sim \operatorname{Rng} f$ is infinite. □

BIBLIOGRAPHY

1. Malcev, A. I. *Algorithms and Recursive Functions*. Groningen: Wolters-Noordhoff (1970).
2. Rogers, H. *Theory of Recursive Functions and Effective Computability*. New York: McGraw-Hill (1967).
3. Smullyan, R. M. *Theory of Formal Systems*. Princeton: Princeton University Press (1961).

EXERCISES

6.29. Let $f\colon \omega \to \omega$. Then the following conditions are equivalent:

(1) f is recursive;
(2) $\{(x, fx) : x \in \omega\}$ is an r.e. relation;
(3) $\{(x, fx) : x \in \omega\}$ is a recursive relation.

6.30. Prove that the class of r.e. sets is closed under union and intersection using the argument following 6.8, but rigorously.

6.31. Show that if A is a Σ_n-set, $n > 0$, and f is partial recursive, then f^*A is Σ_n.

6.32. If A and B are r.e. sets, then there exist r.e. sets C, D such that $C \subseteq A$, $D \subseteq B$, $C \cup D = A \cup B$, and $C \cap D = 0$.

6.33. Suppose that f and g are unary recursive functions, g is one-one, $\operatorname{Rng} g$ is recursive, and $\forall x(fx \geq gx)$. Show that $\operatorname{Rng} f$ is recursive.

6.34. For each of the following determine if the set in question is recursive, r.e., or has an r.e. complement:

(1) $\{x$: there are at least x consecutive 7's in the decimal representation of $\pi\}$;
(2) $\{x$: there is a run of exactly x consecutive 7's in the decimal representation of $\pi\}$;
(3) $\{x : \varphi_x^1$ is total$\}$;
(4) $\{x$: Dmn φ_x^1 is recursive$\}$.

6.35. There are $\aleph_0$ r.e. sets which are not recursive.

6.36. There is a recursive set A such that $\bigcap_{x \in A}$ Dmn φ_x^1 is not r.e.

6.37. If A is productive, then so is $\{e$: Dmn $\varphi_e^1 \subseteq A\}$.

6.38. There are $2^{\aleph_0}$ productive sets. *Hint:* Let $A = \{e$: Dmn $\varphi_e^1 \subseteq \omega \sim \mathrm{K}\}$. Show that $A \subseteq \omega \sim \mathrm{K}$, $(\omega \sim \mathrm{K}) \sim A$ is infinite, and any set P with $A \subseteq P \subseteq \omega \sim \mathrm{K}$ is productive.

6.39. Any infinite r.e. set is the disjoint union of a creative set and a productive set. *Hint:* say Rng $f = A$. Let $gn = f\mu i(fi \neq gj$ for all $j < n)$. Show that $g^*\mathrm{K}$ is creative and $A \sim g^*\mathrm{K}$ is productive.

6.40. If B is r.e. and $A \cap B$ is productive, then A is productive.

6.41. There is an r.e. set which is neither recursive, simple, nor creative. *Hint:* let A be simple and set $B = \{x : (x)_0 \in A\}$.

6.42. For $A \subseteq \omega$ the following are equivalent:

(1) A is recursive and $A \neq 0$;
(2) there is a recursive function f with Rng $f = A$ and $\forall x \in \omega(fx \leq f(x+1))$.

6.43. For $A \subseteq \omega$ the following are equivalent:

(1) A is productive;
(2) there is a partial recursive function f such that $\forall e \in \omega$ (if Dmn $\varphi_e^1 \subseteq A$ then fe is defined and $fe \in A \sim$ Dmn φ_e^1).

6.44. If A is creative, B is r.e., and $A \cap B = 0$, then $A \cup B$ is creative.

6.45. There is a set A such that both A and $\omega \sim A$ are productive.

6.46. If A is productive and B is simple, then $A \cap B$ is productive.

6.47. Two sets A and B are *strongly recursively inseparable* if $A \cap B = 0$, $\omega \sim (A \cup B)$ is infinite, and for every r.e. set C, $C \sim A$ infinite $\Rightarrow C \cap B \neq 0$, $C \sim B$ infinite $\Rightarrow C \cap A \neq 0$. Show that if A and B are r.e. but strongly recursively inseparable, then:

(1) A and B are recursively inseparable.
(2) $A \cup B$ is simple.
(3) neither A nor B is creative.
(4) A and B are not effectively inseparable.

6.48. Show that there exist two r.e. strongly recursively inseparable sets. *Hint:* let $E = \{(e, x) : \exists y((e, x, y) \in \mathrm{T}_1)\}$. Show that there exist recursive functions f, g such that

$$E = \{(fi, gi) : i < \omega\}.$$

Show that there exist recursive functions h, k such that

$$\begin{aligned}
h0 &= \mu i(gi > 3fi);\\
k0 &= \mu i(gi > 3fi \text{ and } gi \neq gh0);\\
h(n+1) &= \mu i(gi > 3fi \mathbin{\&} \forall j \leq n(gi \neq gkj) \mathbin{\&}\\
&\qquad \forall j \leq n(fi \neq fhj) \mathbin{\&} \forall j \leq n(gi \neq ghj));\\
k(n+1) &= \mu i(gi > 3fi \mathbin{\&} \forall j \leq n+1(gi \neq ghj) \mathbin{\&}\\
&\qquad \forall j \leq n(fi \neq fkj) \mathbin{\&} \forall j \leq n(gi \neq gkj)).
\end{aligned}$$

Let $A = \operatorname{Rng}(g \circ h)$, $B = \operatorname{Rng}(g \circ k)$.

7 Survey of Recursion Theory

We have developed recursion theory as much as we need for our later purposes in logic. But in this chapter we want to survey, without proofs, some further topics. Most of these topics are also frequently useful in logical investigations.

Turing Degrees

Let g be a function mapping ω into ω. Imagine a Turing machine equipped with an oracle—an inpenetrable black box—which gives the answer gx when presented with x. The function g may be nonrecursive, so that the oracle is not an effective device. Rigorously, one defines a g-Turing machine just like Turing machines were defined in 1.1, except that $v_1, \ldots, v_{2m}$ are arbitrary members of $\{0, 1, 2, 3, 4, 5\}$. And one adds one more stipulation in 1.2:

> If $w = 5$, and $F(e - 1) = 0$ or $Fe = 1$, then $F' = F$, $d' = f$, $e' = e$, while if $w = 5$ and $0\ 1^{(x+1)}\ 0$ lies on F ending at e, then $0\ 1^{(x+1)}\ 0\ 1^{(gx+1)}\ 0$ lies on F' ending at e', $e' = e + gx + 2$, F' is otherwise like F and $d' = f$.

Then the notion of *g-Turing computable* function is easily defined.

One can also define *g-recursive function*: in 3.1, each class A is required to have g as a member. These two notions, g-Turing computable and g-recursive function, are shown equivalent just as in Chapter 3. In fact, most considerations of Chapters 1 through 6 carry over to this situation. If h is g-recursive, we also say that h is *recursive in g*. One can extend the notion in an obvious way to a set of F of functions, arriving at the notion of a function being recursive in F. At present we restrict ourselves to the simpler notion. We say that h and g are *Turing equivalent* if each is recursive in the other.

This establishes an equivalence relation on the set of all functions mapping ω into ω. The equivalence classes are called *Turing degrees of unsolvability*. Each equivalence class has at most $\aleph_0$ members (actually exactly $\aleph_0$, as is easily seen), since there are only $\aleph_0$ possible Turing machines with oracles. Clearly then there are exp $\aleph_0$ degrees. Let D be the set of degrees. For α, $\beta \in D$ we write $\alpha \leq \beta$ provided there exist $f \in \alpha$ and $g \in \beta$ with f recursive in g. This relation $\leq$ makes D into a partially ordered set. Clearly the degree of recursive functions, denoted by 0, is the least element of D.

Many of the important results about D are concerned with trying to describe the partial ordering $\leq$. A complete description is far from being known. The rather scattered results which we now want to mention are among the strongest facts known. Some of their proofs are quite complicated, involving *priority* arguments, a kind of argument seemingly unique to this area.

Proposition 7.1. *Any two elements of D have a least upper bound.*

Theorem 7.2. *There exist two elements of D without a greatest lower bound.*

Theorem 7.3. *In D, no ascending sequence* $\alpha_0 < \alpha_1 < \cdots$ *has a least upper bound.*

Proposition 7.4. *Every element of D has only countably many predecessors.*

An element α of D is *minimal*, if $0 < \alpha$ and there is no β with $0 < \beta < \alpha$.

Theorem 7.5. *There are* exp $\aleph_0$ *minimal degrees.*

A subset E of D is an *initial segment of* D provided that for all $\alpha, \beta \in D$, if $\alpha < \beta \in E$, then $\alpha \in E$.

Theorem 7.6. *Any finite distributive lattice can be embedded as an initial segment of D; likewise any countable Boolean algebra and any countable ordinal.*

One of the main open problems in the theory of degrees is the conjecture that every finite lattice can be embedded as an initial segment in D.

There are some special degrees of particular importance for applications to logic. A degree α is *recursively enumerable* (r.e.) provided that $\chi_A \in \alpha$ for some r.e. set A. Note that there are only $\aleph_0$ r.e. degrees. We let $0'$ be the degree of χ_K. We know from Theorem 6.19, p. 98 that $\chi_B \in 0'$ for any creative set B; and $0'$ is the largest r.e. degree.

Theorem 7.7. *No r.e. degree is minimal.*

Theorem 7.8. *There are two minimal degrees with join* $0'$.

Corollary 7.9. *There are degrees $\leq 0'$ which are not r.e.*

Theorem 7.10. *For every nonzero r.e. degree α there is a minimal degree $\leq \alpha$.*

Partial Recursive Functionals

A *partial functional* is a function F such that for some $m, n \in \omega$, the domain of F is a subset of ${}^m({}^\omega\omega) \times {}^n\omega$, while the range of F is a subset of ω; additionally we assume $m + n > 0$. In case $m = 0$ we are dealing with the partial functions of Chapter 5. In case the domain of F is all of ${}^m({}^\omega\omega) \times {}^n\omega$, we call F *total.* An *(m, n)-relation* R is any subset of ${}^m({}^\omega\omega) \times {}^n\omega$. We now wish to give a reasonable meaning to F and R being recursive, and to R being recursively enumerable. Since a function cannot be presented in its entirety to a machine it is natural to seek a definition of these notions in which only initial segments of functions are given. If $\mathfrak{A} = (f_0, \ldots, f_{m-1}, x_0, \ldots, x_{n-1}) \in {}^m({}^\omega\omega) \times {}^n\omega$, we let for any $y \in \omega$

$$\tilde{\mathfrak{A}}y = (\tilde{f}_0 y, \ldots, \tilde{f}_{n-1} y, x_0, \ldots, x_{n-1}) \in {}^{m+n}\omega.$$

Now we say that an (m, n)-relation R is *recursively enumerable* (r.e.) provided that there is an $(m + n + 1)$-ary recursive relation $S \subseteq {}^{m+n+1}\omega$ such that for all $\mathfrak{A} \in {}^m({}^\omega\omega) \times {}^n\omega$,

$$\mathfrak{A} \in R \qquad \text{iff } \exists x \in \omega[(\tilde{\mathfrak{A}}x, x) \in S].$$

Obviously this definition coincides with the definition of r.e. relation if $m = 0$. The definition is motivated as follows. We generate the members of S one after the other. Having generated a member $(y_0, \ldots, y_{m-1}, z_0, \ldots, z_{n-1}, x)$ of S, we have implicitly generated each member $\mathfrak{A}$ of R such that $\tilde{\mathfrak{A}}x = (y_0, \ldots, y_{m-1}, z_0, \ldots, z_{n-1})$. Eventually each member of R is generated in this fashion. A partial functional F is *partial recursive* provided that its *graph*

$$R = \{(\mathfrak{A}, x) : \mathfrak{A} \in \text{Dmn } F, F\mathfrak{A} = x\}$$

is r.e. Again this notion coincides with the old definition for $m = 0$. Given $\mathfrak{A} \in \text{Dmn } F$, clearly the above generation of R constitutes an effective calculation of $F\mathfrak{A}$ (provided there is some way to recognize effectively that $\tilde{\mathfrak{A}}x = (y_0, \ldots, y_{m-1}, z_0, \ldots, z_{n-1})$ for given $(y_0, \ldots, y_{m-1}, z_0, \ldots, z_{n-1})$). An (m, n)-relation R is *recursive* provided χ_R is recursive. These definitions form the basis for a generalized recursion theory. This generalization, expounded at length in Shoenfield [9], has many of the properties of ordinary recursion theory; the enumeration, iteration, and recursion theorems carry over, as well as the considerations concerning the arithmetical hierarchy. As is suggested above, there is a strong connection between generalized recursion theory and relative recursiveness:

Theorem 7.11. *A function $f\colon \omega \to \omega$ is recursive in a function $g\colon \omega \to \omega$ iff there is a total recursive functional $F\colon {}^\omega\omega \times \omega \to \omega$ such that for all $x \in \omega$, $fx = F(g, x)$.*

The notion of a functional also enables one to clarify the role of the sets Δ_n in the arithmetical hierarchy:

Theorem 7.12. *A relation is Δ_{n+1} iff it is recursive in $\{\chi_A: A \text{ is } \Pi_n\}$ iff it is recursive in $\{\chi_A: A \text{ is } \Sigma_n\}$.*

The notion of recursive functionals also makes possible the construction of a new hierarchy. An (m, n)-relation R is Σ_m^1 (resp. Π_m^1) where $m \geq 1$ provided there is a recursive relation S so that for all $\mathfrak{A} \in {}^m({}^\omega\omega) \times {}^n\omega$ we have

$$\mathfrak{A} \in R \qquad \text{iff } Q_1 \cdots Q_{m+1}[(\mathfrak{A}, \mathfrak{B}) \in S],$$

where $Q_1, \ldots, Q_m$ are quantifiers $\forall$ or $\exists$ on functions (members of ${}^\omega\omega$), alternately $\forall$ and $\exists$ (with $Q_1 = \exists$ (resp. $Q_1 = \forall$), while Q_{m+1} is a quantifier $\forall x$ or $\exists x$ on numbers. By collapsing quantifiers, it is easy to see that any second-order prefix can be put in this form (see Chapter 30). The classification of relations in the sets Σ_m^1 and Π_m^1 forms the *analytical hierarchy*. Again we set $\Delta_m^1 = \Sigma_m^1 \cap \Pi_m^1$. The theory of this hierarchy shows considerable similarity, in results and proofs, to the classical topological theory of analytic sets. For example, we have

Theorem 7.13 *If P and Q are disjoint Σ_1^1 relations, then there is a Δ_1^1 relation R such that $P \subseteq R$ and $Q \subseteq {\sim}R$.*

Isols

Two sets $A, B \subseteq \omega$ are said to be *recursively equivalent* if there is a one-one partial recursive function f such that $A \subseteq \operatorname{Dmn} f$ and $f^*A = B$. This establishes an equivalence relation on the set of all subsets of ω; the equivalence classes are called *recursive equivalence types* (RET's). They are the effective version of cardinal numbers.

Proposition 7.14. *If α and β are* RET's, *then there exist $A \in \alpha$ and $B \in \beta$ such that A and B are recursively separable.*

Proposition 7.15. *If α and β are* RET's, *A, $A' \in \alpha$, B, $B' \in \beta$, A and B are recursively separable, and A' and B' are recursively separable, then $A \cup B$ is recursively equivalent to $A' \cup B'$.*

By 7.14 and 7.15, we can define a binary operation $+$ on RET by setting $\alpha + \beta =$ recursive equivalence type of $A \cup B$, where $A \in \alpha$, $B \in \beta$, and A and B are recursively separable.

Recall the function J_2 from 3.60. It is a one-one function mapping $\omega \times \omega$ onto ω.

Proposition 7.16. *If $\alpha, \beta \in$ RET, A, $A' \in \alpha$, and B, $B' \in \beta$, then $J_2^*(A \times B)$ is recursively equivalent to $J_2^*(A' \times B')$.*

It follows that we can define $\cdot$ on RET by setting $\alpha \cdot \beta$ = recursive equivalence type of $J_2^*(A \times B)$, where $A \in \alpha$ and $B \in \beta$.

Proposition 7.17. *Addition and multiplication of* RET's *are commutative and associative. Multiplication is distributive over addition.*

The structure (RET, $+$, $\cdot$) is not, however, a ring, and it cannot be embedded in a ring. This can be seen for example, from the fact that $\alpha + \beta = \alpha$ where α and β are respectively the recursive equivalence types of ω and of 1. Since $\beta + \beta \neq \beta$, β is not the additive zero of (RET, $+$, $\cdot$), so this structure cannot even be embedded in a ring.

For each $n \in \omega$, let $\bar{n}$ be the recursive equivalence type of n. Then $^-$ is an isomorphic embedding of (ω, $+$, $\cdot$) into (RET, $+$, $\cdot$).

The structure (RET, $+$, $\cdot$) has a simple substructure which is much more closely related to (ω, $+$, $\cdot$). To define it, let us call a set $A \subseteq \omega$ *isolated* if it is not recursively equivalent to any proper subset $B \subset A$. An RET α is an *isol* if it has an isolated member. We denote by ISOL the collection of all isol's.

Theorem 7.18. ISOL *is closed under* $+$ *and* $\cdot$. *For any* $\alpha, \beta, \gamma \in$ ISOL *we have:*

(*i*) $\alpha + \beta = \alpha + \gamma$ *implies* $\beta = \gamma$;
(*ii*) $\alpha \cdot \beta = \alpha \cdot \gamma$ *and* $\alpha \neq \bar{0}$ *imply* $\beta = \gamma$;
(*iii*) (ω, $+$, $\cdot$) *is a substructure of* (ISOL, $+$, $\cdot$).

The structure (ISOL, $+$, $\cdot$) can be embedded in a ring ISOL*, which has the ordinary ring of integers as a substructure. It has many interesting properties. Since it has zero divisors, it cannot be embedded in a field.

Recursive Real Numbers

It is natural to try to effectivize common notions of mathematics, such as the notion of a real number. We give here a few of the relevant definitions and results.

Let $\mathbb{Q}$ be the set of rational numbers. A sequence $r \in {}^\omega\mathbb{Q}$ is *recursive* iff there exist unary recursive functions f, g, h such that for all $n \in \omega$,

$$rn = (fn - gn)/(1 + hn).$$

Thus if $\varepsilon n = 1/2^n$ for all $n \in \omega$, then ε is recursive. In fact we may take $fn = 1$ for all n, $gn = 0$ for all n, and $hn = 2^n - 1$ for all n. Now a recursive sequence $r \in {}^\omega\mathbb{Q}$ *recursively converges* to a real number α provided there is a unary recursive function k such that for all $n \in \omega$ and all $n \geq km$ we have $|rn - \alpha| < \varepsilon m$. A real number α is *recursive* if there is a recursive sequence of rationals which recursively converges to α.

Theorem 7.19. *The set of recursive real numbers forms a subfield F of the field of real numbers. Every rational number is recursive. F is countable.*

There is a Cauchy recursive sequence of rationals which does not converge recursively.

Theorem 7.20. *If $r \in {}^{\omega}\mathbb{Q}$ is recursive, strictly monotone, converges to a recursive real number α, then r recursively converges to α.*

A sequence $r \in {}^{\omega}F$ is *recursive* provided there are binary recursive functions f, g, h, k such that for all $m, n \in \omega$ and all $p \geq k(m, n)$ we have

$$|r_n - \{[f(p, n) - g(p, n)]/[1 + h(p, n)]\}| < \epsilon m.$$

Many other concepts of ordinary mathematics can be given effective formulations in a similar way.

Word Problem for Groups

There is a classical problem in group theory which has been given a negative solution using notions of recursive function theory. We shall give a precise formulation of it. Let X be a nonempty set. We form the *free group generated by* X as follows. For each $x \in X$ let $x' = (X, x)$. Note that $'$ is a one-one function whose range is disjoint from X. A finite sequence (perhaps 0) of elements of $X \cup \text{Rng}'$ is called a *word* on X; we let W_X be the set of all words on X. Let $\equiv$ be the smallest equivalence relation on W_X containing all pairs $(0, aa')$ and $(0, a'a)$ with $a \in X$. It is easily seen that if $a, b, c, d \in W_X$, $a \equiv b$, and $c \equiv d$, then $ac \equiv bd$. Hence there is a binary operation $\cdot$ on the set F_X of equivalence classes under $\equiv$ such that $[a]\cdot[b] = [ab]$ for all a, $b \in W_X$. Under this operation F_X becomes a group, called the *free group generated by* X. A *defining relation over* X is a pair (a, b) of words over X. If R is a set of defining relations over X, we let R^* be the normal subgroup of F_X generated by all elements $[a]\cdot[b]^{-1}$ with $(a, b) \in R$. Let $F_{X,R} = F_X/R^*$. A group G is *determined* by generators X and defining relations R if it is isomorphic to $F_{X,R}$; then (X, R) is a *presentation* of G. It is easily seen that every group has a presentation. If X and R are finite, then (X, R) is a *finite presentation* and G is *finitely presentable*. If f is a one-one map of $X \cup X'$ into ω, then any word x of W_X can be given a Gödel number $g_f x$ by

$$g_f x = \prod_{i<m} p_i^{fxi+1}$$

where x is of length m. We say that the word problem for (X, R) is *recursively solvable* provided that for some such f,

$$\{(g_f a, g_f b) : h[a] = h[b]\}$$

is recursive, where h is the natural homomorphism of F_X onto $F_{X,R}$. For X and R finite, this definition does not depend on the choice of f.

Theorem 7.21 (Novikov). *There is a group G with a finite presentation (X, R) which is recursively unsolvable. Thus there is no automatic procedure for determining of a pair of words (a, b) whether they become equal upon applying the relations in R.*

Solvability of Diophantine Equations.

A *diophantine equation* is an equation of the form $P(x_0, \ldots, x_{m-1}) = 0$, where $P(x_0, \ldots, x_{m-1})$ is a polynomial in indeterminants $x_0, \ldots, x_{m-1}$ with integer coefficients. A classical problem of number theory, called *Hilbert's tenth problem* (see Davis [2]) is whether there is an automatic method for determining whether an arbitrary diophantine equation has an integral solution. By means of Gödel numbering this question can be given a rigorous form. The answer (Theorem 7.24) is negative, and follows from an even stronger result which we now want to formulate. An n-ary relation $R \subseteq {}^n\omega$ is called *diophantine* if there is a polynomial $P(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1})$ with integral coefficients such that

$$R = \{x \in {}^n\omega : \text{there exist } y_0, \ldots, y_{m-1} \in \omega \text{ such that } P(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}) = 0\}.$$

Theorem 7.22 (J. Robinson, M. Davis, Y. Matiyasevic). *A relation is r.e. iff it is diophantine.*

As an interesting corollary we have

Theorem 7.23. *There is a polynomial with integral coefficients such that its positive values, when members of ω are substituted, are exactly all positive primes.*

It is also easy to derive the solution to Hilbert's tenth problem from 7.22:

Theorem 7.24. *There is no automatic method which, presented with a diophantine equation ε, will decide whether ε has a solution.*

BIBLIOGRAPHY

1. Boone, W. W. The word problem. *Ann. Math.*, *70* (1959), 207–265.
2. Davis, M. Hilbert's tenth problem is unsolvable. *Amer. Math. Monthly*, *80* (1973), 233–269.
3. Dekker, J. C. E. *Les Fonctions Combinatoires et les Isols*. Paris: Gauthier-Villars (1966).
4. Dekker, J. C. E. and Myhill, J. Recursive equivalence types. *Univ. Calif. Publ. Math.*, *3* (1960), 67–214.
5. Hermes, H. *Enumerability, Decidability, Computability*. New York: Springer (1969).