Herbert B. Enderton

A MATHEMATICAL INTRODUCTION TO LOGIC

SECOND EDITION

A Mathematical
Introduction to Logic

This Page Intentionally Left Blank

A Mathematical Introduction to Logic

Second Edition

Herbert B. Enderton
University of California, Los Angeles

HAP HARCOURT ACADEMIC PRESS

A Harcourt Science and Technology Company

San Diego   New York   Boston
London   Toronto   Sydney   Tokyo

for Eric and Bert

This Page Intentionally Left Blank

Contents

Preface

This book, like the first edition, presents the basic concepts and results of logic: the topics are proofs, truth, and computability. As before, the presentation is directed toward the reader with some mathematical background and interests. In this revised edition, in addition to numerous "local" changes, there are three "global" ways in which the presentation has been changed:

First, I have attempted to make the material more *accessible* to the typical undergraduate student. In the main development, I have tried not to take for granted information or insights that might be unavailable to a junior-level mathematics student.

Second, for the instructor who wants to fit the book to his or her course, the organization has been made more *flexible*. Footnotes at the beginning of many of the sections indicate optional paths the instructor — or the independent reader — might choose to take.

Third, theoretical *computer science* has influenced logic in recent years, and some of that influence is reflected in this edition. Issues of computability are taken more seriously. Some material on finite models has been incorporated into the text.

The book is intended to serve as a textbook for an introductory mathematics course in logic at the junior-senior level. The objectives are to present the important concepts and theorems of logic and to explain their significance and their relationship to the reader's other mathematical work.

As a text, the book can be used in courses anywhere from a quarter to a year in length. In one quarter, I generally reach the material on models of first-order theories (Section 2.6). The extra time afforded by a semester would permit some glimpse of undecidability, as in Section 3.0. In a second

term, the material of Chapter 3 (on undecidability) can be more adequately covered.

The book is intended for the reader who has not studied logic previously, but who has some experience in mathematical reasoning. There are no specific prerequisites aside from a willingness to function at a certain level of abstraction and rigor. There is the inevitable use of basic set theory. Chapter 0 gives a concise summary of the set theory used. One should not begin the book by studying this chapter; it is instead intended for reference if and when the need arises. The instructor can adjust the amount of set theory employed; for example it is possible to avoid cardinal numbers completely (at the cost of losing some theorems). The book contains some examples drawn from abstract algebra. But they are just examples, and are not essential to the exposition. The later chapters (Chapter 3 and 4) tend to be more demanding of the reader than are the earlier chapters.

Induction and recursion are given a more extensive discussion (in Section 1.4) than has been customary. I prefer to give an informal account of these subjects in lectures and have a precise version in the book rather than to have the situation reversed.

Exercises are given at the end of nearly all the sections. If the exercise bears a boldface numeral, then the results of that exercise are used in the exposition in the text. Unusually challenging exercises are marked with an asterisk.

I cheerfully acknowledge my debt to my teachers, a category in which I include also those who have been my colleagues or students. I would be pleased to receive comments and corrections from the users of this book. The Web site for the book can be found at http://www.math.ucla.edu/~hbe/amil.

Introduction
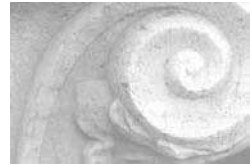
Symbolic logic is a mathematical model of deductive thought. Or at least that was true originally; as with other branches of mathematics it has grown beyond the circumstances of its birth. Symbolic logic is a model in much the same way that modern probability theory is a model for situations involving chance and uncertainty.

How are models constructed? You begin with a real-life object, for example an airplane. Then you select some features of this original object to be represented in the model, for example its shape, and others to be ignored, for example its size. And then you build an object that is like the original in some ways (which you call essential) and unlike it in others (which you call irrelevant). Whether or not the resulting model meets its intended purpose will depend largely on the selection of the properties of the original object to be represented in the model.

Logic is more abstract than airplanes. The real-life objects are certain "logically correct" deductions. For example,

> All men are mortal.
> Socrates is a man.
> Therefore, Socrates is mortal.

The validity of inferring the third sentence (the conclusion) from the first two (the assumptions) does not depend on special idiosyncrasies of Socrates. The inference is justified by the form of the sentences rather than by empirical facts about mortality. It is not really important here what "mortal" means; it does matter what "all" means.

> Borogoves are mimsy whenever it is brillig.
> It is now brillig, and this thing is a borogove.
> Hence this thing is mimsy.

Again we can recognize that the third sentence follows from the first two, even without the slightest idea of what a mimsy borogove might look like.

Logically correct deductions are of more interest than the above frivolous examples might suggest. In fact, axiomatic mathematics consists of many such deductions laid end to end. These deductions made by the working mathematician constitute real-life originals whose features are to be mirrored in our model.

The logical correctness of these deductions is due to their form but is independent of their content. This criterion is vague, but it is just this sort of vagueness that prompts us to turn to mathematical models. A major goal will be to give, within a model, a precise version of this criterion. The questions (about our model) we will initially be most concerned with are these:

1. What does it mean for one sentence to "follow logically" from certain others?
2. If a sentence does follow logically from certain others, what methods of *proof* might be necessary to establish this fact?
3. Is there a gap between what we can *prove* in an axiomatic system (say for the natural numbers) and what is *true* about the natural numbers?
4. What is the connection between logic and computability?

Actually we will present two models. The first (sentential logic) will be very simple and will be woefully inadequate for interesting deductions. Its inadequacy stems from the fact that it preserves only some crude properties of real-life deductions. The second model (first-order logic) is admirably suited to deductions encountered in mathematics. When a working mathematician asserts that a particular sentence follows from the axioms of set theory, he or she means that this deduction can be translated to one in our model.

This emphasis on mathematics has guided the choice of topics to include. This book does not venture into many-valued logic, modal logic, or intuitionistic logic, which represent different selections of properties of real-life deductions.

Thus far we have avoided giving away much information about what our model, first-order logic, is like. As brief hints, we now give some examples of the expressiveness of its formal language. First, take the English sentence that asserts the set-theoretic principle of extensionality, "If the same things are members of a first object as are members of a second object, then those objects are the same." This can be translated into our first-order language as

$$\forall x \, \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y).$$

As a second example, we can translate the sentence familiar to calculus students, "For every positive number ε there is a positive number δ such that for any x whose distance from a is less than δ, the distance between $f(x)$ and b is less than ε" as

$$\forall \varepsilon (\varepsilon > 0 \rightarrow \exists \delta (\delta > 0 \land \forall x (dxa < \delta \rightarrow dfxb < \varepsilon))).$$

We have given some hints as to what we intend to do in this book. We should also correct some possible misimpressions by saying what we are not going to do. This book does not propose to teach the reader how to think. The word "logic" is sometimes used to refer to remedial thinking, but not by us. The reader already knows how to think. Here are some intriguing concepts to think about.

Chapter
Z E R O

Useful Facts about Sets

We assume that the reader already has some familiarity with normal everyday set-theoretic apparatus. Nonetheless, we give here a brief summary of facts from set theory we will need; this will at least serve to establish the notation. It is suggested that the reader, instead of poring over this chapter at the outset, simply refer to it if and when issues of a set-theoretic nature arise in later chapters. The author's favorite book on set theory is of course his *Elements of Set Theory* (see the list of references at the end of this book).

First a word about jargon. Throughout the book we will utilize an assortment of standard mathematical abbreviations. We use "⊣" to signify the end of a proof. A sentence "If ..., then ..." will sometimes be abbreviated "... ⇒ ...." We also have "⇐" for the converse implication (for the peculiar way the word "implication" is used in mathematics). For "if and only if" we use the shorter "iff" (this has become part of the mathematical language) and the symbol "⇔." For the word "therefore" we have the "∴" abbreviation.

The notational device that extracts "$x \neq y$" as the denial of "$x = y$" and "$x \notin y$" as the denial of "$x \in y$" will be extended to other cases. For example, in Section 1.2 we define "$\Sigma \models \tau$"; then "$\Sigma \not\models \tau$" is its denial.

Now then, a *set* is a collection of things, called its members or elements. As usual, we write "$t \in A$" to say that t is a member of A, and "$t \notin A$" to say that t is not a member of A. We write "$x = y$" to

mean that x and y are the same object. That is, the expression "x" on the left of the equals sign is a name for the same object as is named by the other expression "y." If $A = B$, then for any object t it is automatically true that $t \in A$ iff $t \in B$. This holds simply because A and B are the same thing. The converse is the principle of extensionality: If A and B are sets such that for every object t,

$$t \in A \quad \text{iff} \quad t \in B,$$

then $A = B$. This reflects the idea of what a set *is*; a set is determined just by its members.

A useful operation is that of adjoining one extra object to a set. For a set A, let $A; t$ be the set whose members are (i) the members of A, plus (ii) the (possibly new) member t. Here t may or may not already belong to A, and we have

$$A; t = A \cup \{t\}$$

using notation defined later, and

$$t \in A \quad \text{iff} \quad A; t = A.$$

One special set is the empty set $\varnothing$, which has no members at all. Any other set is said to be *nonempty*. For any object x there is the singleton set $\{x\}$ whose only member is x. More generally, for any finite number $x_1, \ldots, x_n$ of objects there is the set $\{x_1, \ldots, x_n\}$ whose members are exactly those objects. Observe that $\{x, y\} = \{y, x\}$, as both sets have exactly the same members. We have only used different expressions to denote the set. If order matters, we can use ordered pairs (discussed later).

This notation will be stretched to cover some simple infinite cases. For example, $\{0, 1, 2, \ldots\}$ is the set $\mathbb{N}$ of natural numbers, and $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$ is the set $\mathbb{Z}$ of all integers.

We write "$\{x \mid \_x\_\}$" for the set of all objects x such that $\_x\_$. We will take considerable liberty with this notation. For example, $\{\langle m, n \rangle \mid m < n \text{ in } \mathbb{N}\}$ is the set of all ordered pairs of natural numbers for which the first component is smaller than the second. And $\{x \in A \mid \_x\_\}$ is the set of all elements x in A such that $\_x\_$.

If A is a set all of whose members are also members of B, then A is a subset of B, abbreviated "$A \subseteq B$." Note that any set is a subset of itself. Also, $\varnothing$ is a subset of every set. ("$\varnothing \subseteq A$" is "vacuously true," since the task of verifying, for every member of $\varnothing$, that it also belongs to A requires doing nothing at all. Or from another point of view, "$A \subseteq B$" can be false only if some member of A fails to belong to B. If $A = \varnothing$, this is impossible.) From the set A we can form a new set, the *power set* $\mathcal{P}A$ of A, whose members are the subsets of A. Thus

$$\mathcal{P}A = \{x \mid x \subseteq A\}.$$

For example,

$$\mathcal{P}\varnothing = \{\varnothing\},$$
$$\mathcal{P}\{\varnothing\} = \{\varnothing, \{\varnothing\}\}.$$

The *union* of A and B, $A \cup B$, is the set of all things that are members of A or B (or both). For example, $A; t = A \cup \{t\}$. Similarly, the *intersection* of A and B, $A \cap B$, is the set of all things that are members of both A and B. Sets A and B are *disjoint* iff their intersection is empty (i.e., if they have no members in common). A collection of sets is *pairwise disjoint* iff any two members of the collection are disjoint.

More generally, consider a set A whose members are themselves sets. The union, $\bigcup A$, of A is the set obtained by dumping all the members of A into a single set:

$$\bigcup A = \{x \mid x \quad \text{belongs to some member of } A\}.$$

Similarly for nonempty A,

$$\bigcap A = \{x \mid x \quad \text{belongs to all members of } A\}.$$

For example, if

$$A = \{\{0, 1, 5\}, \{1, 6\}, \{1, 5\}\},$$

then

$$\bigcup A = \{0, 1, 5, 6\},$$

$$\bigcap A = \{1\}.$$

Two other examples are

$$A \cup B = \bigcup\{A, B\},$$

$$\bigcup \mathcal{P}A = A.$$

In cases where we have a set A_n for each natural number n, the union of all these sets, $\bigcup\{A_n \mid n \in \mathbb{N}\}$, is usually denoted "$\bigcup_{n \in \mathbb{N}} A_n$" or just "$\bigcup_n A_n$."

The ordered pair $\langle x, y \rangle$ of objects x and y must be defined in such a way that

$$\langle x, y \rangle = \langle u, v \rangle \quad \text{iff} \quad x = u \quad \text{and} \quad y = v.$$

Any definition that has this property will do; the standard one is

$$\langle x, y \rangle = \{\{x\}, \{x, y\}\}.$$

For ordered triples we define

$$\langle x, y, z \rangle = \langle\langle x, y \rangle, z \rangle.$$

More generally we define n-tuples recursively by

$$\langle x_1, \ldots, x_{n+1} \rangle = \langle \langle x_1, \ldots, x_n \rangle, x_{n+1} \rangle$$

for $n > 1$. It is convenient to define also $\langle x \rangle = x$; the preceding equation then holds also for $n = 1$. S is a *finite sequence* (or *string*) of members of A iff for some positive integer n, we have $S = \langle x_1, \ldots, x_n \rangle$, where each $x_i \in A$. (Finite sequences are often defined to be certain finite functions, but the above definition is slightly more convenient for us.)

A *segment* of the finite sequence $S = \langle x_1, \ldots, x_n \rangle$ is a finite sequence

$$\langle x_k, x_{k+1}, \ldots, x_{m-1}, x_m \rangle, \qquad \text{where} \quad 1 \leq k \leq m \leq n.$$

This segment is an *initial segment* iff $k = 1$ and it is *proper* iff it is different from S.

If $\langle x_1, \ldots, x_n \rangle = \langle y_1, \ldots, y_n \rangle$, then it is easy to see that $x_i = y_i$ for $1 \leq i \leq n$. (The proof uses induction on n and the basic property of ordered pairs.) But if $\langle x_1, \ldots, x_m \rangle = \langle y_1, \ldots, y_n \rangle$, then it does not in general follow that $m = n$. After all, every ordered triple is also an ordered pair. But we claim that m and n can be unequal only if some x_i is itself a finite sequence of y_j's, or the other way around:

> **LEMMA 0A**   Assume that $\langle x_1, \ldots, x_m \rangle = \langle y_1, \ldots, y_m, \ldots, y_{m+k} \rangle$. Then $x_1 = \langle y_1, \ldots, y_{k+1} \rangle$.

> PROOF.   We use induction on m. If $m = 1$, the conclusion is immediate. For the inductive step, assume that $\langle x_1, \ldots, x_m, x_{m+1} \rangle = \langle y_1, \ldots, y_{m+k}, y_{m+1+k} \rangle$. Then the first components of this ordered pair must be equal: $\langle x_1, \ldots, x_m \rangle = \langle y_1, \ldots, y_{m+k} \rangle$. Now apply the inductive hypothesis.   ⊣

For example, suppose that A is a set such that no member of A is a finite sequence of other members. Then if $\langle x_1, \ldots, x_m \rangle = \langle y_1, \ldots, y_n \rangle$ and each x_i and y_j is in A, then by the above lemma $m = n$. Whereupon we have $x_i = y_i$ as well.

From sets A and B we can form their *Cartesian product*, the set $A \times B$ of all pairs $\langle x, y \rangle$ for which $x \in A$ and $y \in B$. A^n is the set of all n-tuples of members of A. For example, $A^3 = (A \times A) \times A$.

A *relation* R is a set of ordered pairs. For example, the ordering relation on the numbers 0–3 is captured by — and in fact *is* — the set of ordered pairs

$$\{\langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 0, 3 \rangle, \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 3 \rangle\}.$$

The *domain* of R (written dom R) is the set of all objects x such that $\langle x, y \rangle \in R$ for some y. The *range* of R (written ran R) is the set of all objects y such that $\langle x, y \rangle \in R$ for some x. The union of dom R and ran R is the *field* of R, fld R.

An *n-ary relation* on A is a subset of A^n. If $n > 1$, it is a relation. But a 1-ary (unary) relation on A is simply a subset of A. A particularly simple binary relation on A is the equality relation $\{\langle x, x \rangle \mid x \in A\}$ on A. For an n-ary relation R on A and subset B of A, the *restriction* of R to B is the intersection $R \cap B^n$. For example, the relation displayed above is the restriction to the set $B = \{0, 1, 2, 3\}$ of the ordering relation on $\mathbb{N}$.

A *function* is a relation F with the property of being *single-valued*: For each x in dom F there is only one y such that $\langle x, y \rangle \in F$. As usual, this unique y is said to be the value $F(x)$ that F assumes at x. (This notation goes back to Euler. It is a pity he did not choose $(x)F$ instead; that would have been helpful for the *composition* of functions: $f \circ g$ is the function whose value at x is $f(g(x))$, obtained by applying first g and then f.)

We say that F *maps* A into B and write

$$F : A \to B$$

to mean that F is a function, dom $F = A$, and ran $F \subseteq B$. If in addition ran $F = B$, then F maps A *onto* B. F is *one-to-one* iff for each y in ran F there is only one x such that $\langle x, y \rangle \in F$. If the pair $\langle x, y \rangle$ is in dom F, then we let $F(x, y) = F(\langle x, y \rangle)$. This notation is extended to n-tuples; $F(x_1, \ldots, x_n) = F(\langle x_1, \ldots, x_n \rangle)$.

An *n-ary operation* on A is a function mapping A^n into A. For example, addition is a binary operation on $\mathbb{N}$, whereas the successor operation S (where $S(n) = n + 1$) is a unary operation on $\mathbb{N}$. If f is an n-ary operation on A, then the *restriction* of f to a subset B of A is the function g with domain B^n which agrees with f at each point of B^n. Thus,

$$g = f \cap (B^n \times A).$$

This g will be an n-ary operation on B iff B is *closed* under f, in the sense that $f(b_1, \ldots, b_n) \in B$ whenever each b_i is in B. In this case, $g = f \cap B^{n+1}$, in agreement with our definition of the restriction of a relation. For example, the addition operation on $\mathbb{N}$, which contains such triples as $\langle \langle 3, 2 \rangle, 5 \rangle$, is the restriction to $\mathbb{N}$ of the addition operation on $\mathbb{R}$, which contains many more triples.

A particularly simple unary operation on A is the *identity* function Id on A, given by the equation

$$Id(x) = x \qquad \text{for} \quad x \in A.$$

Thus $Id = \{\langle x, x \rangle \mid x \in A\}$.

For a relation R, we define the following:

R is *reflexive* on A iff $\langle x, x \rangle \in R$ for every x in A.

R is *symmetric* iff whenever $\langle x, y \rangle \in R$, then also $\langle y, x \rangle \in R$.

R is *transitive* iff whenever both $\langle x, y \rangle \in R$ and $\langle y, z \rangle \in R$ (if this ever happens), then also $\langle x, z \rangle \in R$.

R satisfies *trichotomy* on A iff for every x and y in A, exactly one of the three possibilities, $\langle x, y \rangle \in R$, $x = y$, or $\langle y, x \rangle \in R$, holds.

R is an *equivalence relation* on A iff R is a binary relation on A that is reflexive on A, symmetric, and transitive.

R is an *ordering relation* on A iff R is transitive and satisfies trichotomy on A.

For an equivalence relation R on A we define, for $x \in A$, the *equivalence class* $[x]$ of x to be $\{y \mid \langle x, y \rangle \in R\}$. The equivalence classes then *partition* A. That is, the equivalence classes are subsets of A such that each member of A belongs to exactly one equivalence class. For x and y in A,

$$[x] = [y] \quad \text{iff} \quad \langle x, y \rangle \in R.$$

The set $\mathbb{N}$ of natural numbers is the set $\{0, 1, 2, \ldots\}$. (Natural numbers can also be defined set-theoretically, a point that arises briefly in Section 3.7.) A set A is *finite* iff there is some one-to-one function f mapping (for some natural number n) the set A onto $\{0, 1, \ldots, n-1\}$. (We can think of f as "counting" the members of A.)

A set A is *countable* iff there is some function mapping A one-to-one into $\mathbb{N}$. For example, any finite set is obviously countable. Now consider an infinite countable set A. Then from the given function f mapping A one-to-one into $\mathbb{N}$, we can extract a function f' mapping A one-to-one *onto* $\mathbb{N}$. For some $a_0 \in A$, $f(a_0)$ is the least member of ran f, let $f'(a_0) = 0$. In general there is a unique $a_n \in A$ such that $f(a_n)$ is the $(n+1)$st member of ran f; let $f'(a_n) = n$. Note that $A = \{a_0, a_1, \ldots\}$. (We can also think of f' as "counting" the members of A, only now the counting process is infinite.)

THEOREM 0B Let A be a countable set. Then the set of all finite sequences of members of A is also countable.

PROOF.   The set S of all such finite sequences can be characterized by the equation

$$S = \bigcup_{n \in \mathbb{N}} A^{n+1}.$$

Since A is countable, we have a function f mapping A one-to-one into $\mathbb{N}$.

The basic idea is to map S one-to-one into $\mathbb{N}$ by assigning to $\langle a_0, a_1, \ldots, a_m \rangle$ the number $2^{f(a_0)+1} 3^{f(a_1)+1} \cdot \ldots \cdot p_m^{f(a_m)+1}$, where p_m is the $(m+1)$st prime. This suffers from the defect that this assignment might not be well-defined. For conceivably there could be $\langle a_0, a_1, \ldots, a_m \rangle = \langle b_0, b_1, \ldots, b_n \rangle$, with a_i and b_j in A but with $m \neq n$. But this is not serious; just assign to each member of S the *smallest* number obtainable in the above

fashion. This gives us a well-defined map; it is easy to see that it is one-to-one.   ⊣

At times we will speak of *trees*, which can be useful in providing intuitive pictures of some situations. But our comments on trees will always be informal; the theorems and proofs will not rely on trees. Accordingly, our discussion here of trees will be informal.

For each tree there is an underlying finite partial ordering. We can draw a picture of this partial ordering R; if $\langle a, b \rangle \in R$, then we put a lower than b and connect the points by a line. Pictures of two typical tree orderings are shown.
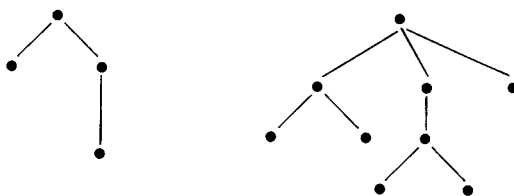
(In mathematics, trees grow downward, not upward.) There is always a highest point in the picture (the *root*). Furthermore, while branching is permitted below some vertex, the points *above* any given vertex must lie along a line.

In addition to this underlying finite partial ordering, a tree also has a labeling function whose domain is the set of vertices. For example, one tree, in which the labels are natural numbers, is shown.
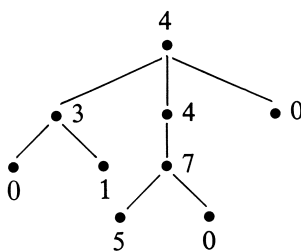
At a few points in the book we will use the axiom of choice. But usually these uses can be eliminated if the theorems in question are restricted to countable languages. Of the many equivalent statements of the axiom of choice, Zorn's lemma is especially useful.

Say that a collection C of sets is a *chain* iff for any elements x and y of C, either $x \subseteq y$ or $y \subseteq x$.

ZORN'S LEMMA   Let A be a set such that for any chain $C \subseteq A$, the set $\bigcup C$ is in A. Then there is some element $m \in A$ which is maximal in the sense that it is not a subset of any other element of A.

Cardinal Numbers

All infinite sets are big, but some are bigger than others. (For example, the set of real numbers is bigger than the set of integers.) Cardinal numbers provide a convenient, although not indispensable, way of talking about the size of sets.

It is natural to say that two sets A and B have the same size iff there is a function that maps A one-to-one onto B. If A and B are finite, then this concept is equivalent to the usual one: If you count the members of A and the members of B, then you get the same number both times. But it is applicable even to infinite sets A and B, where counting is difficult.

Formally, then, say that A and B are *equinumerous* (written $A \sim B$) iff there is a one-to-one function mapping A onto B. For example, the set $\mathbb{N}$ of natural numbers and the set $\mathbb{Z}$ of integers are equinumerous. It is easy to see that equinumerosity is reflexive, symmetric, and transitive.

For finite sets we can use natural numbers as measures of size. The same natural number would be assigned to two finite sets (as measures of their size) iff the sets were equinumerous. Cardinal numbers are introduced to enable us to generalize this situation to infinite sets.

To each set A we can assign a certain object, the *cardinal number* (or *cardinality*) of A (written card A), in such a way that two sets are assigned the same cardinality iff they are equinumerous:

$$\text{card } A = \text{card } B \quad \text{iff} \quad A \sim B. \tag{K}$$

There are several ways of accomplishing this; the standard one these days takes card A to be the least ordinal equinumerous with A. (The success of this definition relies on the axiom of choice.) We will not discuss ordinals here, since for our purposes it matters very little what card A actually is, any more than it matters what the number 2 actually is. What matters most is that (K) holds. It is helpful, however, if for a finite set A, card A is the natural number telling how many elements A has. Something is a *cardinal number*, or simply a *cardinal*, iff it is card A for some set A.

(Georg Cantor, who first introduced the concept of cardinal number, characterized in 1895 the cardinal number of a set M as "the general concept which, with the help of our active intelligence, comes from the set M upon abstraction from the nature of its various elements and from the order of their being given.")

Say that A is *dominated* by B (written $A \preceq B$) iff A is equinumerous with a subset of B. In other words, $A \preceq B$ iff there is a one-to-one function mapping A into B. The companion concept for cardinals is

$$\text{card } A \leq \text{card } B \quad \text{iff} \quad A \preceq B.$$

(It is easy to see that $\leq$ is well defined; that is, whether or not $\kappa \leq \lambda$ depends only on the cardinals κ and λ themselves, and not the choice of

sets having these cardinalities.) Dominance is reflexive and transitive. A set A is dominated by $\mathbb{N}$ iff A is countable. The following is a standard result in this subject.

> **SCHRÖDER–BERNSTEIN THEOREM**   (a) For any sets A and B, if $A \preceq B$ and $B \preceq A$, then $A \sim B$.
> (b) For any cardinal numbers κ and λ, if $\kappa \leq \lambda$ and $\lambda \leq \kappa$, then $\kappa = \lambda$.

Part (b) is a simple restatement of part (a) in terms of cardinal numbers. The following theorem, which happens to be equivalent to the axiom of choice, is stated in the same dual manner.

> **THEOREM 0C**   (a) For any sets A and B, either $A \preceq B$ or $B \preceq A$.
> (b) For any cardinal numbers κ and λ, either $\kappa \leq \lambda$ or $\lambda \leq \kappa$.

Thus of any two cardinals, one is smaller than the other. (In fact, any nonempty set of cardinal numbers contains a smallest member.) The smallest cardinals are those of finite sets: $0, 1, 2, \ldots$. There is next the smallest infinite cardinal, card $\mathbb{N}$, which is given the name $\aleph_0$. Thus we have

$$0, 1, 2, \ldots, \aleph_0, \aleph_1, \ldots,$$

where $\aleph_1$ is the smallest cardinal larger than $\aleph_0$. The cardinality of the real numbers, card $\mathbb{R}$, is called "$2^{\aleph_0}$." Since $\mathbb{R}$ is uncountable, we have $\aleph_0 < 2^{\aleph_0}$.

The operations of addition and multiplication, long familiar for finite cardinals, can be extended to all cardinals. To compute $\kappa + \lambda$ we choose disjoint sets A and B of cardinality κ and λ, respectively. Then

$$\kappa + \lambda = \operatorname{card}(A \cup B).$$

This is well defined; i.e., $\kappa + \lambda$ depends only on κ and λ, and not on the choice of the disjoint sets A and B. For multiplication we use

$$\kappa \cdot \lambda = \operatorname{card}(A \times B).$$

Clearly these definitions are correct for finite cardinals. The arithmetic of infinite cardinals is surprisingly simple (with the axiom of choice). The sum or product of two infinite cardinals is simply the larger of them:

> **CARDINAL ARITHMETIC THEOREM**   For cardinal numbers κ and λ, if $\kappa \leq \lambda$ and λ is infinite, then $\kappa + \lambda = \lambda$. Furthermore, if $\kappa \neq 0$, then $\kappa \cdot \lambda = \lambda$.

In particular, for infinite cardinals κ,

$$\aleph_0 \cdot \kappa = \kappa.$$

THEOREM 0D   For an infinite set A, the set $\bigcup_n A^{n+1}$ of all finite sequences of elements of A has cardinality equal to card A.

We already proved this for the case of a countable A (see Theorem 0B).

PROOF.   Each A^{n+1} has cardinality equal to card A, by the cardinal arithmetic theorem (applied n times). So we have the union of $\aleph_0$ sets of this size, yielding $\aleph_0 \cdot$ card $A =$ card A points altogether.
$$\dashv$$

EXAMPLE.   It follows that the set of algebraic numbers has cardinality $\aleph_0$. First, we can identify each polynomial (in one variable) over the integers with the sequence of its coefficients. Then by the theorem there are $\aleph_0$ polynomials. Each polynomial has a finite number of roots. To give an extravagant upper bound, note that even if each polynomial had $\aleph_0$ roots, we would then have $\aleph_0 \cdot \aleph_0 = \aleph_0$ algebraic numbers altogether. Since there are at least this many, we are done.

Since there are uncountably many (in fact, $2^{\aleph_0}$) real numbers, it follows that there are uncountably many (in fact, $2^{\aleph_0}$) transcendental numbers.

Chapter ONE

Sentential Logic

SECTION 1.0
Informal Remarks on Formal Languages

We are about to construct (in the next section) a language into which we can translate English sentences. Unlike natural languages (such as English or Chinese), it will be a formal language, with precise formation rules. But before the precision begins, we will consider here some of the features to be incorporated into the language.

As a first example, the English sentence "Traces of potassium were observed" can be translated into the formal language as, say, the symbol **K**. Then for the closely related sentence "Traces of potassium were not observed," we can use $(\neg \mathbf{K})$. Here $\neg$ is our negation symbol, read as "not." One might also think of translating "Traces of potassium were not observed" by some new symbol, for example, **J**, but we will prefer instead to break such a sentence down into atomic parts as much as possible. For an unrelated sentence, "The sample contained chlorine," we choose, say, the symbol **C**. Then the following compound English sentences can be translated as the formulas shown at the right:

If traces of potassium were observed, then the sample did not contain chlorine.

$(\mathbf{K} \rightarrow (\neg \mathbf{C}))$

The sample contained chlorine, and traces of potassium were observed.

$(\mathbf{C} \wedge \mathbf{K})$

The second case uses our conjunction symbol $\wedge$ to translate "and." The first one uses the more familiar arrow to translate "if ... , then ...." In the following example the disjunction symbol $\vee$ is used to translate "or":

Either no traces of potassium were observed, or the sample did not contain chlorine.	$((\neg\,\mathbf{K}) \vee (\neg\,\mathbf{C}))$
Neither did the sample contain chlorine, nor were traces of potassium observed.	$(\neg\,(\mathbf{C} \vee \mathbf{K}))$ or $((\neg\,\mathbf{C}) \wedge (\neg\,\mathbf{K}))$

For this last sentence, we have two alternative translations. The relationship between them will be discussed later.

One important aspect of the decompositions we will make of compound sentences is that whenever we are given the truth or falsity of the atomic parts, we can then immediately calculate the truth or falsity of the compound. Suppose, for example, that the chemist emerges from her laboratory and announces that she observed traces of potassium but that the sample contained no chlorine. We then know that the four above sentences are true, false, true, and false, respectively. In fact, we can construct in advance a table giving the four possible experimental results (Table I). We will return to the discussion of such tables in Section 1.2.

TABLE I

K	C	$(\neg(\mathbf{C} \vee \mathbf{K}))$	$((\neg\,\mathbf{C}) \wedge (\neg\,\mathbf{K}))$
F	*F*	*T*	*T*
F	*T*	*F*	*F*
T	*F*	*F*	*F*
T	*T*	*F*	*F*

Use of formal languages will allow us to escape from the imprecision and ambiguities of natural languages. But this is not done without cost; our formal languages will have a sharply limited degree of expressiveness.

In order to describe a formal language we will generally give three pieces of information:

1. We will specify the set of symbols (the alphabet). In the present case of sentential logic some of the symbols are

$$(,), \rightarrow, \neg, \mathbf{A}_1, \mathbf{A}_2, \ldots .$$

2. We will specify the rules for forming the "grammatically correct" finite sequences of symbols. (Such sequences will be called *well-formed formulas* or *wffs*.) For example, in the present case

$$(\mathbf{A}_1 \rightarrow (\neg\,\mathbf{A}_2))$$

will be a wff, whereas

$$)) \rightarrow \mathbf{A}_3$$

will not.

3. We will also indicate the allowable translations between English and the formal language. The symbols $\mathbf{A}_1, \mathbf{A}_2, \ldots$ can be translations of declarative English sentences.

Only in this third part is any meaning assigned to the wffs. This process of assigning meaning guides and motivates all we do. But it will also be observed that it would theoretically be possible to carry out various manipulations with wffs in complete ignorance of any possible meaning. A person aware of only the first two pieces of information listed above could perform some of the things we will do, but it would make no sense to him.

Before proceeding, let us look briefly at another class of formal languages of widespread interest today. These are the languages used by (or at least in connection with) digital computers.

There are many of these languages. In one of them a typical wff is

$$01101011010100011111000100001111010.$$

In another a typical wff is

$$\text{STEP\#ADDIMAX, A.}$$

(Here # is a symbol called a blank; it is brought into the alphabet so that a wff will be a string of symbols.) A well-known language called C++ has wffs such as

$$\text{while}(*s{+}{+});$$

In all cases there is a given way of translating the wffs into English, and (for a restricted class of English sentences) a way of translating from English into the formal language. But the computer is unaware of the English language. An unthinking automaton, the computer juggles symbols and follows its program slavishly. We could approach formal languages that way, too, but it would not be much fun.

SECTION 1.1

The Language of Sentential Logic

We assume we are given an infinite sequence of distinct objects which we will call symbols, and to which we now give names (Table II). We further assume that no one of these symbols is a finite sequence of other symbols.

TABLE II

Symbol	Verbose name	Remarks
(	left parenthesis	punctuation
)	right parenthesis	punctuation
¬	negation symbol	English: not
∧	conjunction symbol	English: and
∨	disjunction symbol	English: or (inclusive)
→	conditional symbol	English: if __, then __
↔	biconditional symbol	English: if and only if
A_1	first sentence symbol	
A_2	second sentence symbol	
. . .		
A_n	nth sentence symbol	
. . .		

Several remarks are now in order:

1. The five symbols

$$\neg, \ \wedge, \ \vee, \ \rightarrow, \ \leftrightarrow$$

are called *sentential connective* symbols. Their use is suggested by the English translation given above. The sentential connective symbols, together with the parentheses, are the *logical symbols*. In translating to and from English, their role never changes. The sentence symbols are the *parameters* (or *nonlogical symbols*). Their translation is not fixed; instead they will be open to a variety of interpretations, as we shall illustrate shortly.

2. We have included infinitely many sentence symbols. On the one hand, a more modest alternative would be to have *one* sentence symbol A, and a prime $'$. Then we could use the potentially infinite sequence

$$A, A', A'', \ldots$$

in place of

$$A_1, A_2, A_3 \ldots.$$

This alternative has the advantage that it brings the total number of different symbols down to nine. On the other hand, a *less* modest alternative would be to allow an arbitrary set of set sentence symbols, countable or not. Much of what is said in this chapter would continue to be applicable in this case; the exceptions are primarily in Section 1.7.

3. Some logicians prefer to call A_n the nth *proposition* symbol (and to speak of propositional logic instead of sentential logic). This stems

from wanting the word "sentence" to refer to a particular utterance, and wanting a proposition to be that which a sentence asserts.

4. We call these objects "symbols," but we remain neutral as to what the exact ontological status of symbols might be. In the leftmost column of our list of symbols, names of the symbols are printed. For example, $\mathbf{A}_{243}$ is one symbol, namely the 243rd sentence symbol. (On the other hand, "$\mathbf{A}_{243}$" is a name of that symbol. The conditional symbol may or may not have the geometric property of being shaped like an arrow, although its name "$\rightarrow$" does.) The symbols themselves can be sets, numbers, marbles, or objects from a universe of linguistic objects. In the last case, it is conceivable that they are actually the same things as the names we use for them. Another possibility, which will be explained in the next chapter, is that the sentence symbols are themselves formulas in another language.

5. We have assumed that no symbol is a finite sequence of other symbols. We mean by this that not only are the symbols listed distinct (e.g., $\mathbf{A}_3 \neq \leftrightarrow$), but no one of them is a finite sequence of two or more symbols. For example, we demand that $\mathbf{A}_3 \neq \langle \neg, \mathbf{A}_4, (\rangle$. The purpose of this assumption is to assure that finite sequences of symbols are uniquely decomposable. If

$$\langle a_1, \ldots, a_m \rangle = \langle b_1, \ldots, b_n \rangle$$

and each a_i and each b_j is a symbol, then $m = n$ and $a_i = b_i$. (See Chapter 0, Lemma 0A, and subsequent remarks.)

An *expression* is a finite sequence of symbols. We can specify an expression by concatenating the names of the symbols; thus $(\neg \mathbf{A}_1)$ is the sequence $\langle (, \neg, \mathbf{A}_1,) \rangle$. This notation is extended: If α and β are sequences of symbols, then $\alpha\beta$ is the sequence consisting first of the symbols in the sequence α followed by the symbols in the sequence β.

For example, if α and β are the expressions given by the equations

$$\alpha = (\neg \mathbf{A}_1),$$

$$\beta = \mathbf{A}_2,$$

then $(\alpha \rightarrow \beta)$ is the expression

$$((\neg \mathbf{A}_1) \rightarrow \mathbf{A}_2).$$

We should now look at a few examples of possible translations of English sentences into expressions of the formal language. Let $\mathbf{A}$, $\mathbf{B}$, $\ldots$, $\mathbf{Z}$ be the first twenty-six sentence symbols. (For example, $\mathbf{E} = \mathbf{A}_5$.)

1. English: The suspect must be released from custody. Translation: $\mathbf{R}$.
   English: The evidence obtained is admissible. Translation: $\mathbf{E}$.
   English: The evidence obtained is inadmissible. Translation: $(\neg \mathbf{E})$.

English: The evidence obtained is admissible, and the suspect need not be released from custody. Translation: $(\mathbf{E} \wedge (\neg \mathbf{R}))$.

English: Either the evidence obtained is admissible, or the suspect must be released from custody (or possibly both). Translation: $(\mathbf{E} \vee \mathbf{R})$.

English: Either the evidence obtained is admissible, or the suspect must be released from custody, but not both. Translation: $((\mathbf{E} \vee \mathbf{R}) \wedge (\neg (\mathbf{E} \wedge \mathbf{R})))$. We intend always to use the symbol $\vee$ to translate the word "or" in its inclusive meaning of "and/or."

English: The evidence obtained is inadmissible, but the suspect need not be released from custody. Translation: $((\neg \mathbf{E}) \wedge (\neg \mathbf{R}))$. On the other hand, the expression $((\neg \mathbf{E}) \vee (\neg \mathbf{R}))$ translates the English: Either the evidence obtained is inadmissible or the suspect need not be released from custody.

2. English: If wishes are horses, then beggars will ride. Translation: $(\mathbf{W} \rightarrow \mathbf{B})$.

English: Beggars will ride if and only if wishes are horses. Translation: $(\mathbf{B} \leftrightarrow \mathbf{W})$.

3. English: This commodity constitutes wealth if and only if it is transferable, limited in supply, and either productive of pleasure or preventive of pain. Translation: $(\mathbf{W} \leftrightarrow (\mathbf{T} \wedge (\mathbf{L} \wedge (\mathbf{P} \vee \mathbf{Q}))))$. Here $\mathbf{W}$ is the translation of "This commodity constitutes wealth." Of course in the preceding example we used $\mathbf{W}$ to translate a different sentence. We are not tied to any one translation.

One note of caution: Do not confuse a sentence in the English language (Roses are red) with a translation of that sentence in the formal language (such as $\mathbf{R}$). These are different. The English sentence is presumably either true or false. But the formal expression is simply a sequence of symbols. It may indeed be interpreted in one context as a true (or false) English sentence, but it can have other interpretations in other contexts.

Now some expressions cannot be obtained as translations of any English sentences but are mere nonsense, such as

$$((\rightarrow \mathbf{A}_3.$$

We want to define the well-formed formulas (wffs) to be the "grammatically correct" expressions; the nonsensical expressions must be excluded. The definition will have the following consequences:

(a) Every sentence symbol is a wff.

(b) If α and β are wffs, then so are $(\neg \alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$, and $(\alpha \leftrightarrow \beta)$.

(c) No expression is a wff unless it is compelled to be one by (a) and (b).

We want to make this third property (about compulsion) more precise. A *well-formed formula* (or simply *formula* or *wff*) is an expression that can be built up from the sentence symbols by applying some finite number of times the *formula-building operations* (on expressions) defined by the equations

$$\mathcal{E}_{\neg}(\alpha) = (\neg\,\alpha)$$
$$\mathcal{E}_{\wedge}(\alpha, \beta) = (\alpha \wedge \beta),$$
$$\mathcal{E}_{\vee}(\alpha, \beta) = (\alpha \vee \beta),$$
$$\mathcal{E}_{\rightarrow}(\alpha, \beta) = (\alpha \rightarrow \beta),$$
$$\mathcal{E}_{\leftrightarrow}(\alpha, \beta) = (\alpha \leftrightarrow \beta).$$

For example,

$$((A_1 \wedge A_{10}) \rightarrow ((\neg A_3) \vee (A_8 \leftrightarrow A_3)))$$

is a wff, as can be seen by contemplating its ancestral tree:
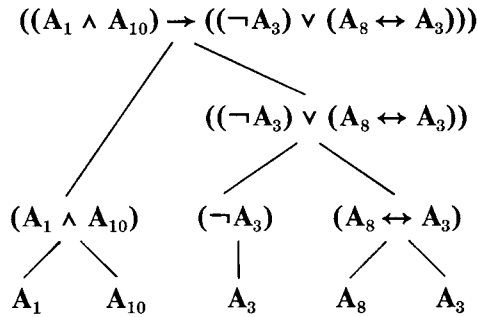
The tree illustrates the construction of the expression from four sentence symbols by five applications of formula-building operations. This example is atypical in that it uses all five formula-building operations. For a smaller example, A_3 is a wff; its ancestral tree has only a single vertex; each formula-building operation is used zero times. But this is as small as examples get; we do not count the empty sequence as being "built up from the sentence symbols."

This sort of construction, of taking some basic building blocks (here the sentence symbols) and "closing" under some operations (here five operations), occurs frequently both in logic and in other branches of mathematics. In Section 1.4, we will examine this sort of construction in a more general setting.

We can elaborate on the "building" idea as follows. Define a *construction sequence* to be a finite sequence $\langle \varepsilon_1, \ldots, \varepsilon_n \rangle$ of expressions

such that for each $i \leq n$ we have at least one of

$$\varepsilon_i \text{ is a sentence symbol}$$
$$\varepsilon_i = \mathcal{E}_\neg(\varepsilon_j) \text{ for some } j < i$$
$$\varepsilon_i = \mathcal{E}_\square(\varepsilon_j, \varepsilon_k) \text{ for some } j < i, k < i$$

where $\square$ is one of the binary connectives, $\wedge, \vee, \rightarrow, \leftrightarrow$. Then the well-formed formulas can be characterized as the expressions α such that some construction sequence ends with α. We can think of ε_i as the expression at stage i in the building process.

For our earlier example,

$$((\mathbf{A}_1 \wedge \mathbf{A}_{10}) \rightarrow ((\neg \mathbf{A}_3) \vee (\mathbf{A}_8 \leftrightarrow \mathbf{A}_3)))$$

we obtain a construction sequence by squashing its ancestral tree into a linear ordering.

One feature of this sort of construction is that it yields an *induction principle*. Say that a set S is *closed* under a two-place function f iff whenever $x \in S$ and $y \in S$ then $f(x, y) \in S$, and similarly for one-place functions and so forth.

> **INDUCTION PRINCIPLE**   If S is a set of wffs containing all the sentence symbols and closed under all five formula-building operations, then S is the set of *all* wffs.

> FIRST PROOF.   Consider an arbitrary wff α. It is built up from sentence symbols by applying some finite number of times the formula-building operations. Working our way up the corresponding ancestral tree, we find that each expression in the tree belongs to S. Eventually (that is, after a finite number of steps) at the top of the tree we find that $\alpha \in S$. $\dashv$

> SECOND PROOF.   We will repeat the argument, but without the trees. Consider an arbitrary wff α. It is the last member of some construction sequence $\langle \varepsilon_1, \ldots, \varepsilon_n \rangle$. By ordinary strong numerical induction on the number i, we see that each $\varepsilon_i \in S, i \leq n$.
>
> That is, we suppose, as our inductive hypothesis, that $\varepsilon_j \in S$ for all $j < i$. We then verify that $\varepsilon_i \in S$, by considering the various cases. So by strong induction on i, it follows that $\varepsilon_i \in S$ for each $i \leq n$. In particular, the last member α belongs to S. $\dashv$

This principle will receive much use in the coming pages. In the following example we use it to show that certain expressions are *not* wffs.

> **EXAMPLE.**   Any expression with more left parentheses than right parentheses is not a wff.

PROOF.   The idea is that we start with sentence symbols (having zero left parentheses and zero right parentheses), and then apply formula-building operations which add parentheses only in matched pairs. We can rephrase this argument as follows: The set of "balanced" wffs (having equal numbers of left and right parentheses) contains all sentence symbols and is closed under the formula-building operations. The induction principle then assures us that all wffs are balanced.   ⊣

A special feature of our particular formula-building operations is that they build *up* and never *down*. That is, the expression $\mathcal{E}_\square(\alpha, \beta)$ always includes as a segment the entire sequence α (and the entire sequence β) plus other symbols. In particular, it is longer than either α or β.

This special feature will simplify the problem of determining, given a wff φ, exactly *how* it was built up. All the building blocks, so to speak, are included as segments in the sequence φ. For example, if φ does not contain the symbol $\mathbf{A}_4$, then φ can be built up without ever using $\mathbf{A}_4$. (See Exercise 4.)

Exercises

1. Give three sentences in English together with translations into our formal language. The sentences should be chosen so as to have an interesting structure, and the translations should each contain 15 or more symbols.

2. Show that there are no wffs of length 2, 3, or 6, but that any other positive length is possible.

3. Let α be a wff; let c be the number of places at which binary connective symbols ($\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$) occur in α; let s be the number of places at which sentence symbols occur in α. (For example, if α is $(\mathbf{A} \rightarrow (\neg \mathbf{A}))$ then $c = 1$ and $s = 2$.) Show by using the induction principle that $s = c + 1$.

4. Assume we have a construction sequence ending in φ, where φ does not contain the symbol $\mathbf{A}_4$. Suppose we delete all the expressions in the construction sequence that contain $\mathbf{A}_4$. Show that the result is still a legal construction sequence.

5. Suppose that α is a wff not containing the negation symbol $\neg$.
   (a) Show that the length of α (i.e., the number of symbols in the string) is odd.
   (b) Show that more than a quarter of the symbols are sentence symbols.
   Suggestion: Apply induction to show that the length is of the form $4k + 1$ and the number of sentence symbols is $k + 1$.

SECTION 1.2

Truth Assignments

We want to define what it means for one wff of our language to fol-
low logically from other wffs. For example, $\mathbf{A}_1$ should follow from
$(\mathbf{A}_1 \wedge \mathbf{A}_2)$. For no matter how the parameters $\mathbf{A}_1$ and $\mathbf{A}_2$ are translated
back into English, if the translation of $(\mathbf{A}_1 \wedge \mathbf{A}_2)$ is true, then the transla-
tion of $\mathbf{A}_1$ must be true. But the notion of all possible translations back
into English is unworkably vague. Luckily the spirit of this notion can
be expressed in a simple and precise way.

Fix once and for all a set $\{F, T\}$ of *truth values* consisting of two
distinct points:

$$F, \quad \text{called } \textit{falsity},$$
$$T, \quad \text{called } \textit{truth}.$$

(It makes no difference what these points themselves are; they might as
well be the numbers 0 and 1.) Then a *truth assignment* v for a set $\mathcal{S}$ of
sentence symbols is a function

$$v : \mathcal{S} \to \{F, T\}$$

assigning either T or F to each symbol in $\mathcal{S}$. These truth assignments
will be used in place of the translations into English mentioned in the
preceding paragraph.

(At this point we have committed ourselves to *two-valued* logic. It is
also possible to study three-valued logic, in which case one has a set of
three possible truth values. And then, of course, it is a small additional
step to allow 512 or $\aleph_0$ truth values; or to take as the set of truth values
the unit interval $[0, 1]$ or some other convenient space. A particularly
interesting case is that for which the truth values form something called
a complete Boolean algebra. But it is two-valued logic that has always
had the greatest significance, and we will confine ourselves to this case.)

Let $\bar{\mathcal{S}}$ be the set of wffs that can be built up from $\mathcal{S}$ by the five
formula-building operations. ($\bar{\mathcal{S}}$ can also be characterized as the set of
wffs whose sentence symbols are all in $\mathcal{S}$; see the remarks at the end of
the preceding section.) We want an extension $\bar{v}$ of v,

$$\bar{v} : \bar{\mathcal{S}} \to \{F, T\},$$

that assigns the correct truth value to each wff in $\bar{\mathcal{S}}$. It should meet the
following conditions:

0. For any $A \in \mathcal{S}$, $\bar{v}(A) = v(A)$. (Thus $\bar{v}$ is an extension of v.)
For any α, β in $\bar{\mathcal{S}}$:

1. $\bar{v}((\neg\alpha)) = \begin{cases} T & \text{if} \quad \bar{v}(\alpha) = F, \\ F & \text{otherwise.} \end{cases}$

2. $\bar{v}((\alpha \wedge \beta)) = \begin{cases} T & \text{if} \quad \bar{v}(\alpha) = T \quad \text{and} \quad \bar{v}(\beta) = T, \\ F & \text{otherwise.} \end{cases}$

3. $\bar{v}((\alpha \vee \beta)) = \begin{cases} T & \text{if} \quad \bar{v}(\alpha) = T \quad \text{or} \quad \bar{v}(\beta) = T \text{ (or both)}, \\ F & \text{otherwise.} \end{cases}$

4. $\bar{v}((\alpha \rightarrow \beta)) = \begin{cases} F & \text{if} \quad \bar{v}(\alpha) = T \quad \text{and} \quad \bar{v}(\beta) = F, \\ T & \text{otherwise.} \end{cases}$

5. $\bar{v}((\alpha \leftrightarrow \beta)) = \begin{cases} T & \text{if} \quad \bar{v}(\alpha) = \bar{v}(\beta), \\ F & \text{otherwise.} \end{cases}$

Conditions 1–5 are given in tabular form in Table III. It is at this point that the intended meaning of, for example, the conjunction symbol enters into our formal proceedings. Note especially the intended meaning of $\rightarrow$. Whenever α is assigned F, then $(\alpha \rightarrow \beta)$ is considered "vacuously true" and is assigned the value T. For this and the other connectives, it is certainly possible to question how accurately we have reflected the common meaning in everyday speech of "if . . . , then," "or," etc. But our ultimate concern lies more with mathematical statements than with the subtle nuances of everyday speech.

TABLE III

α	β	$(\neg \alpha)$	$(\alpha \wedge \beta)$	$(\alpha \vee \beta)$	$(\alpha \rightarrow \beta)$	$(\alpha \leftrightarrow \beta)$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

For example, we might translate the English sentence, "If you're telling the truth then I'm a monkey's uncle," by the formula $(V \rightarrow M)$. We assign this formula the value T whenever you are fibbing. In assigning the value T, we are certainly not asserting any causal connection between your veracity and any simian features of my nephews or nieces. The sentence in question is a *conditional* statement. It makes an assertion about my relatives provided a certain *condition* — that you are telling the truth — is met. Whenever that condition fails, the statement is vacuously true.

Very roughly, we can think of a conditional formula $(\alpha \rightarrow \beta)$ as expressing a *promise* that if a certain condition is met (viz., that α is true), then β is true. If the condition α turns out not to be met, then the promise stands unbroken, regardless of β.

As an example of the calculation of $\bar{v}$, let α be the wff

$$((A_2 \rightarrow (A_1 \rightarrow A_6)) \leftrightarrow ((A_2 \wedge A_1) \rightarrow A_6))$$

and let v be the truth assignment for $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_6\}$ such that

$$v(\mathbf{A}_1) = T,$$
$$v(\mathbf{A}_2) = T,$$
$$v(\mathbf{A}_6) = F.$$

We want to compute $\bar{v}(\alpha)$. We can look at the tree which displays the construction of α:
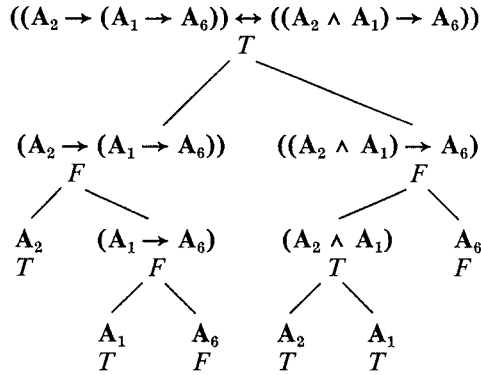
Working from the bottom up, we can assign to each vertex β of the tree the value $\bar{v}(\beta)$. So as a first step we compute

$$\bar{v}((\mathbf{A}_1 \rightarrow \mathbf{A}_6)) = F \quad \text{and} \quad \bar{v}((\mathbf{A}_2 \wedge \mathbf{A}_1)) = T.$$

Next we compute $\bar{v}((\mathbf{A}_2 \rightarrow (\mathbf{A}_1 \rightarrow \mathbf{A}_6))) = F$, and so forth. Finally, at the top of the tree we arrive at $\bar{v}(\alpha) = T$.

Actually this computation can be carried out with much less writing. First, the tree can be given in starker form:
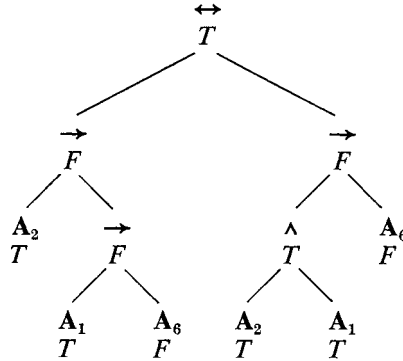
And even this can be compressed into a single line (with the parentheses restored):

$$((A_2 \to (A_1 \to A_6)) \leftrightarrow ((A_2 \wedge A_1) \to A_6)).$$
$$T \quad F \quad T \quad F \quad F \quad \quad T \quad \quad T \quad T \quad T \quad \quad F \quad F$$

THEOREM 12A For any truth assignment v for a set $\mathcal{S}$ there is a unique function $\overline{v} : \overline{\mathcal{S}} \to \{F, T\}$ meeting the aforementioned conditions 0–5.

The full proof of this theorem will emerge in the next two sections (1.3 and 1.4). But it should already seem extremely plausible, especially in light of the preceding example. In proving the existence of $\overline{v}$, the crucial issue will, in effect, be the uniqueness of the trees mentioned in the example.

We say that a truth assignment v *satisfies* φ iff $\overline{v}(\varphi) = T$. (Of course for this to happen, the domain of v must contain all sentence symbols in φ.) Now consider a set Σ of wffs (thought of as hypotheses) and another wff τ (thought of as a possible conclusion).

DEFINITION. Σ *tautologically implies* τ (written $\Sigma \models \tau$) iff every truth assignment for the sentence symbols in Σ and τ that satisfies every member of Σ also satisfies τ.

This definition reflects our intuitive feeling that a conclusion follows from a set of hypotheses if the assumption that the hypotheses are true guarantees that the conclusion is true.

Several special cases of the concept of tautological implication deserve mention. First, take the special case in which Σ is the empty set $\varnothing$. Observe that it is vacuously true that any truth assignment satisfies every member of $\varnothing$. (How could this fail? Only if there was some unsatisfied member of $\varnothing$, which is absurd.) Hence we are left with: $\varnothing \models \tau$ iff every truth assignment (for the sentence symbols in τ) satisfies τ. In this case we say that τ is a *tautology* (written $\models \tau$). For example, the wff $((A_2 \to (A_1 \to A_6)) \leftrightarrow ((A_2 \wedge A_1) \to A_6))$ was in a recent example found to be satisfied by one of the eight possible truth assignments for $\{A_1, A_2, A_6\}$. In fact, it is satisfied by the other seven as well, and hence is a tautology.

Another special case is that in which no truth assignment satisfies every member of Σ. Then for any τ, it is vacuously true that $\Sigma \models \tau$. For example,

$$\{A, (\neg A)\} \models B.$$

There is no deep principle involved here; it is simply a by-product of our definitions.

EXAMPLE. $\{A, (A \to B)\} \models B$. There are four truth assignments for $\{A, B\}$. It is easy to check that only one of these four satisfies both A and $(A \to B)$, namely the v for which $v(A) = v(B) = T$. This v also satisfies B.

If Σ is singleton $\{\sigma\}$, then we write "$\sigma \models \tau$" in place of "$\{\sigma\} \models \tau$." If both $\sigma \models \tau$ and $\tau \models \sigma$, then σ and τ are said to be *tautologically equivalent* (written $\sigma \models\!\!=\!\!\mid \tau$). For example, in Section 1.0 we encountered the wffs $(\neg\,(\mathbf{C}\vee\mathbf{K}))$ and $((\neg\,\mathbf{C})\wedge(\neg\,\mathbf{K}))$ as alternative translations of an English sentence. We can now assert that they are tautologically equivalent.

We can already state here a nontrivial fact whose proof will be given later (in Section 1.7).

> **COMPACTNESS THEOREM**   Let Σ be an infinite set of wffs such that for any finite subset Σ_0 of Σ, there is a truth assignment that satisfies every member of Σ_0. Then there is a truth assignment that satisfies every member of Σ.

This theorem can be restated more simply as: If every finite subset of Σ is satisfiable, then Σ itself is satisfiable. (Readers familiar with some general topology might try to discover why this is called "compactness theorem"; it does assert the compactness of a certain topological space. They can then prove the theorem for themselves, by using Tychonoff's theorem on product spaces.)

Truth Tables

There is a systematic procedure, which we will now illustrate, for checking, given wffs $\sigma_1, \ldots, \sigma_k$, and τ, whether or not

$$\{\sigma_1, \ldots, \sigma_k\} \models \tau.$$

In particular (when $k = 0$), the procedure will, given a wff, decide whether or not it is a tautology.

As a first example, we can show that

$$(\neg\,(\mathbf{A} \wedge \mathbf{B})) \models ((\neg\,\mathbf{A}) \vee (\neg\,\mathbf{B})).$$

To do this, we consider all truth assignments for $\{\mathbf{A}, \mathbf{B}\}$. There are four such assignments; in general there are 2^n truth assignments for a set of n sentence symbols. The four can be listed in a table:

A	B
T	*T*
T	*F*
F	*T*
F	*F*

This table can then be expanded to include $(\neg\,(\mathbf{A}\wedge\mathbf{B}))$ and $((\neg\,\mathbf{A})\vee(\neg\,\mathbf{B}))$. For each formula we compute the *T*'s and *F*'s the way described before, writing the truth value under the correct connective (Table IV). (The two leftmost columns of Table IV are actually unnecessary.) We can now see from this table that all those truth assignments satisfying $(\neg\,(\mathbf{A}\wedge\mathbf{B}))$ — and there are three such — also satisfy $((\neg\,\mathbf{A})\vee(\neg\,\mathbf{B}))$.

TABLE IV

A	B	$(\neg (A \wedge B))$	$((\neg A) \vee (\neg B))$
T	T	$F\ T\ T\ T$	$F\ T\ F\ F\ T$
T	F	$T\ T\ F\ F$	$F\ T\ T\ T\ F$
F	T	$T\ F\ F\ T$	$T\ F\ T\ F\ T$
F	F	$T\ F\ F\ F$	$T\ F\ T\ T\ F$

In fact, the converse holds also, and thus

$$(\neg (A \wedge B)) \models\!\!\models ((\neg A) \vee (\neg B)).$$

To show that $(\neg (A \wedge B)) \not\models ((\neg A) \wedge (\neg B))$ we can construct the table as before. But only one line of the table is needed to establish that there is indeed a truth assignment satisfying $(\neg (A \wedge B))$ that fails to satisfy $((\neg A) \wedge (\neg B))$.

The more generally applicable a procedure it is, the less efficient it is likely to be. For example, to show that

$$\models ((A \vee (B \wedge C)) \leftrightarrow ((A \vee B) \wedge (A \vee C))),$$

we could apply the truth-table method. But this requires eight lines (for the eight possible truth assignments for $\{A, B, C\}$). With a little cleverness, the tedium in this particular case can be reduced:

$$((A \vee (B \wedge C)) \leftrightarrow ((A \vee B) \wedge (A \vee C))).$$
$$\begin{array}{llll} T\ T & T\ \ \ T\ T & T\ T\ T \\ F\ F\ F\ F & T\ \ \ F\ F\ F\ F\ F \\ F\ T\ T\ T\ T & T\ \ \ F\ T\ T\ T\ F\ T\ T \end{array}$$

In the first line we assumed only that $v(A) = T$. Since that is enough information to obtain T for the wff, we assume in all later lines that $v(A) = F$. In the second line we assume that $v(B) = F$; this again lets us obtain T for the wff. So we may limit ourselves to the case $v(B) = T$. Since the expression is symmetric in B and C, we may further suppose $v(C) = T$. This leaves only the third line, whereupon we are done.

As an example of the avoidance of a 16-line table, consider the following tautology:

$$((((P \wedge Q) \rightarrow R) \rightarrow S) \rightarrow ((P \rightarrow R) \rightarrow S)).$$
$$\begin{array}{lll} & T & T\ T \\ F\ \ \ T\ \ \ F\ F\ T \\ T\ T\ T\ R\ R\ \bar{R}\ F\ T\ \ \ T\ R\ R\ \bar{R}\ F \end{array}$$

Here in the first line we dispose of the case where $v(S) = T$. In the second line we dispose of the case where either $v(P) = F$ or $v(Q) = F$.

The third line incorporates the two remaining possibilities; here R is the truth value assigned to **R** and $\overline{R}$ is the opposite value.

For the above example it is possible to see directly why it is a tautology. The stronger the antecedent (the expression on the left side), the weaker the conditional. Thus

$$(\mathbf{P} \wedge \mathbf{Q}) \models \mathbf{P},$$
$$(\mathbf{P} \rightarrow \mathbf{R}) \models ((\mathbf{P} \wedge \mathbf{Q}) \rightarrow \mathbf{R}),$$
$$(((\mathbf{P} \wedge \mathbf{Q}) \rightarrow \mathbf{R}) \rightarrow \mathbf{S}) \models ((\mathbf{P} \rightarrow \mathbf{R}) \rightarrow \mathbf{S}).$$

The problem of developing effective procedures that reduce the tedium is important for theorem proving by machine. Some of the programs here may require testing wffs of sentential logic having thousands of sentence symbols. Truth tables are far too cumbersome for anything this large. The problem of developing highly efficient methods is a current area of research in computer science.

Digression. Applying the truth-table method — carried out in full — to a wff with n sentence symbols requires making a table of 2^n lines. The trouble is, as n increases, 2^n grows "exponentially." For example, suppose you can generate the table at the rate of a million lines per second. (We are supposing you are aided by a computer, of course.) Then if $n = 80$, say, you will need "merely" 2^{80} microseconds to form the complete table. How long is that? Converted to years, 2^{80} microseconds is about 38 billion years. For comparison, the age of the universe is around 15 billion years. The conclusion is that 2^{80} microseconds is more time than there has been time!

Is there a faster method? Might there be some general method that, given any wff α with n sentence symbols, will determine whether or not α is a tautology in only $10n^5$ microseconds (or some other function of n that grows like a polynomial instead of growing exponentially)? (For $n = 80$, $10n^5$ microseconds comes to only 9 hours.) The answer to this question is unknown, but it is widely believed that the answer is negative. The question is called the "P versus NP" problem, the most prominent open problem in theoretical computer science today.

A Selected List of Tautologies

1. Associative and commutative laws for $\wedge$, $\vee$, $\leftrightarrow$.
2. Distributive laws:

$$((\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C})) \leftrightarrow ((\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C}))).$$
$$((\mathbf{A} \vee (\mathbf{B} \wedge \mathbf{C})) \leftrightarrow ((\mathbf{A} \vee \mathbf{B}) \wedge (\mathbf{A} \vee \mathbf{C}))).$$

3. Negation:

$$((\neg(\neg \mathbf{A})) \leftrightarrow \mathbf{A}).$$
$$((\neg(\mathbf{A} \rightarrow \mathbf{B})) \leftrightarrow (\mathbf{A} \wedge (\neg \mathbf{B}))).$$
$$((\neg(\mathbf{A} \leftrightarrow \mathbf{B})) \leftrightarrow ((\mathbf{A} \wedge (\neg \mathbf{B})) \vee ((\neg \mathbf{A}) \wedge \mathbf{B}))).$$

De Morgan's laws:

$$((\neg(\mathbf{A} \wedge \mathbf{B})) \leftrightarrow ((\neg \mathbf{A}) \vee (\neg \mathbf{B}))).$$
$$((\neg(\mathbf{A} \vee \mathbf{B})) \leftrightarrow ((\neg \mathbf{A}) \wedge (\neg \mathbf{B}))).$$

4. Other

Excluded middle:	$(\mathbf{A} \vee (\neg \mathbf{A})).$
Contradiction:	$(\neg(\mathbf{A} \wedge (\neg \mathbf{A}))).$
Contraposition:	$((\mathbf{A} \rightarrow \mathbf{B}) \leftrightarrow ((\neg \mathbf{B}) \rightarrow (\neg \mathbf{A}))).$
Exportation:	$(((\mathbf{A} \wedge \mathbf{B}) \rightarrow \mathbf{C}) \leftrightarrow (\mathbf{A} \rightarrow (\mathbf{B} \rightarrow \mathbf{C}))).$

Exercises

1. Show that neither of the following two formulas tautologically implies the other:

$$(\mathbf{A} \leftrightarrow (\mathbf{B} \leftrightarrow \mathbf{C})),$$
$$((\mathbf{A} \wedge (\mathbf{B} \wedge \mathbf{C})) \vee ((\neg \mathbf{A}) \wedge ((\neg \mathbf{B}) \wedge (\neg \mathbf{C})))).$$

   Suggestion: Only two truth assignments are needed, not eight.

2. (a) Is $(((\mathbf{P} \rightarrow \mathbf{Q}) \rightarrow \mathbf{P}) \rightarrow \mathbf{P})$ a tautology?
   (b) Define σ_k recursively as follows: $\sigma_0 = (\mathbf{P} \rightarrow \mathbf{Q})$ and $\sigma_{k+1} = (\sigma_k \rightarrow \mathbf{P})$. For which values of k is σ_k a tautology? (Part (a) corresponds to $k = 2$.)

3. (a) Determine whether or not $((\mathbf{P} \rightarrow \mathbf{Q}) \vee (\mathbf{Q} \rightarrow \mathbf{P}))$ is a tautology.
   (b) Determine whether or not $((\mathbf{P} \wedge \mathbf{Q}) \rightarrow \mathbf{R})$ tautologically implies $((\mathbf{P} \rightarrow \mathbf{R}) \vee (\mathbf{Q} \rightarrow \mathbf{R})).$

4. Show that the following hold:
   (a) $\Sigma; \alpha \models \beta$ iff $\Sigma \models (\alpha \rightarrow \beta).$
   (b) $\alpha \models\mid\models \beta$ iff $\models (\alpha \leftrightarrow \beta).$
   (Recall that $\Sigma; \alpha = \Sigma \cup \{\alpha\}$, the set Σ together with the one possibly new member α.)

5. Prove or refute each of the following assertions:
   (a) If either $\Sigma \models \alpha$ or $\Sigma \models \beta$, then $\Sigma \models (\alpha \vee \beta).$
   (b) If $\Sigma \models (\alpha \vee \beta)$, then either $\Sigma \models \alpha$ or $\Sigma \models \beta.$

6. (a) Show that if v_1 and v_2 are truth assignments which agree on all the sentence symbols in the wff α, then $\bar{v}_1(\alpha) = \bar{v}_2(\alpha)$. Use the induction principle.

(b) Let $\mathcal{S}$ be a set of sentence symbols that includes those in Σ and τ (and possibly more). Show that $\Sigma \models \tau$ iff every truth assignment for $\mathcal{S}$ which satisfies every member of Σ also satisfies τ. (This is an easy consequence of part (a). The point of part (b) is that we do not need to worry about getting the domain of a truth assignment *exactly* perfect, as long as it is big enough. For example, one option would be always to use truth assignments on the set of *all* sentence symbols. The drawback is that these are infinite objects, and there are a great many — uncountably many — of them.)

7. You are in a land inhabited by people who either always tell the truth or always tell falsehoods. You come to a fork in the road and you need to know which fork leads to the capital. There is a local resident there, but he has time only to reply to one yes-or-no question. What one question should you ask so as to learn which fork to take? *Suggestion*: Make a table.

8. (Substitution) Consider a sequence $\alpha_1, \alpha_2, \ldots$ of wffs. For each wff φ let φ^* be the result of replacing the sentence symbol $\mathbf{A}_n$ by α_n, for each n.
   (a) Let v be a truth assignment for the set of all sentence symbols; define u to be the truth assignment for which $u(\mathbf{A}_n) = \bar{v}(\alpha_n)$. Show that $\bar{u}(\varphi) = \bar{v}(\varphi^*)$. Use the induction principle.
   (b) Show that if φ is a tautology, then so is φ^*. (For example, one of our selected tautologies is $((\mathbf{A} \wedge \mathbf{B}) \leftrightarrow (\mathbf{B} \wedge \mathbf{A}))$. From this we can conclude, by substitution, that $((\alpha \wedge \beta) \leftrightarrow (\beta \wedge \alpha))$ is a tautology, for any wffs α and β.)

9. (Duality) Let α be a wff whose only connective symbols are $\wedge$, $\vee$, and $\neg$. Let α^* be the result of interchanging $\wedge$ and $\vee$ and replacing each sentence symbol by its negation. Show that α^* is tautologically equivalent to $(\neg \alpha)$. Use the induction principle.
   Remark: It follows that if $\alpha \models\!\models \beta$ then $\alpha^* \models\!\models \beta^*$.

10. Say that a set Σ_1 of wffs is *equivalent* to a set Σ_2 of wffs iff for any wff α, we have $\Sigma_1 \models \alpha$ iff $\Sigma_2 \models \alpha$. A set Σ is *independent* iff no member of Σ is tautologically implied by the remaining members in Σ. Show that the following hold.
    (a) A finite set of wffs has an independent equivalent subset.
    (b) An infinite set need not have an independent equivalent subset.
    *(c) Let $\Sigma = \{\sigma_0, \sigma_1, \ldots\}$; show that there is an independent equivalent set Σ'. (By part (b), we cannot hope to have $\Sigma' \subseteq \Sigma$ in general.)

11. Show that a truth assignment v satisfies the wff

$$(\cdots (\mathbf{A}_1 \leftrightarrow \mathbf{A}_2) \leftrightarrow \cdots \leftrightarrow \mathbf{A}_n)$$

iff $v(\mathbf{A}_i) = F$ for an even number of i's, $1 \leq i \leq n$. (By the associative law for $\leftrightarrow$, the placement of the parentheses is not crucial.)

12. There are three suspects for a murder: Adams, Brown, and Clark. Adams says "I didn't do it. The victim was an old acquaintance of Brown's. But Clark hated him." Brown states "I didn't do it. I didn't even know the guy. Besides I was out of town all that week." Clark says "I didn't do it. I saw both Adams and Brown downtown with the victim that day; one of them must have done it." Assume that the two innocent men are telling the truth, but that the guilty man might not be. Who did it?

13. An advertisement for a tennis magazine states, "If I'm not playing tennis, I'm watching tennis. And if I'm not watching tennis, I'm reading about tennis." We can assume that the speaker cannot do more than one of these activities at a time. What is the speaker doing? (Translate the given sentences into our formal language; consider the possible truth assignments.)

14. Let $\mathcal{S}$ be the set of all sentence symbols, and assume that $v : \mathcal{S} \to \{F, T\}$ is a truth assignment. Show there is *at most* one extension $\bar{v}$ meeting conditions 0–5 listed at the beginning of this section. (Suppose that $\bar{v}_1$ and $\bar{v}_2$ are both such extensions. Use the induction principle to show that $\bar{v}_1 = \bar{v}_2$.)

15. Of the following three formulas, which tautologically imply which?
    (a) $(\mathbf{A} \leftrightarrow \mathbf{B})$
    (b) $(\neg ((\mathbf{A} \to \mathbf{B}) \to (\neg (\mathbf{B} \to \mathbf{A}))))$
    (c) $(((\neg \mathbf{A}) \vee \mathbf{B}) \wedge (\mathbf{A} \vee (\neg \mathbf{B})))$

SECTION 1.3
A Parsing Algorithm

The purpose of this section is to prove that we have used enough parentheses to eliminate any ambiguity in analyzing wffs. (The existence of the extension $\bar{v}$ of a truth assignment v will hinge on this lack of ambiguity.[1])

It is instructive to consider the result of not having parentheses at all. The resulting ambiguity is illustrated by the wff

$$\mathbf{A_1} \vee \mathbf{A_2} \wedge \mathbf{A_3},$$

---

[1] The reader who has already accepted the existence of $\bar{v}$ may omit almost all of this section. The final subsection, on omitting parentheses, will still be needed.

which can be formed in two ways, corresponding to $((\mathbf{A_1} \vee \mathbf{A_2}) \wedge \mathbf{A_3})$ and to $(\mathbf{A_1} \vee (\mathbf{A_2} \wedge \mathbf{A_3}))$. If $v(\mathbf{A_1}) = T$ and $v(\mathbf{A_3}) = F$, then there is an unresolvable conflict that arises in trying to compute $\overline{v}(\mathbf{A_1} \vee \mathbf{A_2} \wedge \mathbf{A_3})$.

We must show that with our parentheses this type of ambiguity does not arise but that on the contrary each wff is formed in a unique way. There is one sense in which this fact is unimportant: If it failed, we would simply change notation until it was true. For example, instead of building formulas by means of concatenation, we could have used ordered pairs and triples: $\langle \neg, \alpha \rangle$ and $\langle \alpha, \wedge, \beta \rangle$, and so forth. (This is, in fact, a tidy, but untraditional, method.) It would then be immediate that formulas had unique decompositions. But we do not have to resort to this device, and we will now prove that we do not.

> **LEMMA 13A** Every wff has the same number of left as right parentheses.
>
> PROOF.   This was done as an example at the end of Section 1.1.   ⊣
>
> **LEMMA 13B** Any proper initial segment of a wff contains an excess of left parentheses. Thus no proper initial segment of a wff can itself be a wff.
>
> PROOF.   We apply the induction principle to the set S of wffs possessing the desired property (that proper initial segments are left-heavy). A wff consisting of a sentence symbol alone has no proper initial segments and hence is in S vacuously. To verify that S is closed under $\mathcal{E}_\wedge$, consider α and β in S. The proper initial segments of $(\alpha \wedge \beta)$ are the following:
>
> 1. $($.
> 2. $(\alpha_0$, where α_0 is a proper initial segment of α.
> 3. $(\alpha$.
> 4. $(\alpha \wedge$.
> 5. $(\alpha \wedge \beta_0$, where β_0 is a proper initial segment of β.
> 6. $(\alpha \wedge \beta$.
>
> By applying the inductive hypothesis that α and β are in S (in cases 2 and 5), we obtain the desired conclusion. For closure under the other four formula-building operations, the argument is similar.   ⊣

A Parsing Algorithm

We are now going to describe a procedure that, given an expression, both will determine whether or not the expression is a legal wff, and if it is, will construct the tree showing how it was built up from sentence symbols by application of the formula-building operations. Moreover,

we will see that this tree is *uniquely determined* by the wff. It is this last fact that will assure us that we have enough parentheses for an unambiguous notation.

Assume then that we are given an expression. We will construct a tree, with the given expression at the top. Initially it is the only vertex in the tree, but as the procedure progresses, the tree will grow downward from the given expression. For an example, imagine the tree in Section 1.1.

The algorithm consists of the following four steps:

1. If all minimal vertices (the ones at the bottom) have sentence symbols, then the procedure is completed. (The given expression is indeed a wff, and we have constructed its tree.) Otherwise, select a minimal vertex having an expression that is not a sentence symbol. We examine that expression.

2. The first symbol must* be (. If the second symbol is the negation symbol, jump to step 4. Otherwise, go on to step 3.

3. Scan the expression from the left until first reaching (α, where α is a nonempty expression having a balance between left and right parentheses.** Then α is the first of the two constituents. The next symbol must* be $\wedge$, $\vee$, $\rightarrow$, or $\leftrightarrow$. This is the principal connective. The remainder of the expression, β), must* consist of an expression β and a right parenthesis. We extend the tree by creating two new vertices below the present one, with α as the expression at the "left child" vertex, and β as the expression at the "right child" vertex. Return to step 1.

4. The first two symbols are now known to be ($\neg$. The remainder of the expression, β), must* consist of an expression β and a right parenthesis. We extend the tree by creating one new vertex below the present one, with β as the expression at the child vertex. Return to step 1.

Now for some comments in support of the correctness of this algorithm.

First, given any expression, the procedure halts after a finite number of steps. This is because any vertex contains a shorter expression than the one above it, so the depth of the tree is bounded by the length of the given expression.

Secondly, the choices made by the procedure could not have been made differently. For example, in step 3 we arrive at an expression α. We could not use less than α for a constituent, because it would not have a balance between left and right parentheses (as required by Lemma 13A).

---

* If not, then the expression here is not a wff. We reject the given expression as a non-wff, and halt.

** If the end of the expression is reached before finding such an α, then the expression here is not a wff. We reject the given expression as a non-wff, and halt.

We could not use more than α, because that would have the proper initial segment α that was balanced (violating Lemma 13B). Thus α is forced upon us. And then the choice of the principal connective is inevitable. We conclude that this algorithm constructs the *only* possible tree for the given expression.

Thirdly, if the algorithm uses the footnotes to reject the given expression, then that expression could not have been a wff — the rejection is correct. That is because the only possible attempt to make its tree failed.

Finally, if the algorithm does not use the rejection footnotes, then the given expression is indeed a legal wff. That is because we *have* its tree; by working our way up the tree, we find inductively that every vertex has a wff, including the top vertex.

The second remark above lets us conclude that our language has enough parentheses; each wff has a *unique* tree of the sort constructed here. We have "unique readability." And not only does a unique tree *exist* for a wff, we know how to construct it; we can carry out this algorithm, using a sufficiently large piece of paper.

Now to go back to the question of the existence of the extension $\bar{v}$ of a truth assignment v: It is the uniqueness of the trees that is the crucial fact here. For any wff φ there is a unique tree constructing it. By working our way up this tree, assigning a truth value $\bar{v}(\alpha)$ at each vertex α, we can unambiguously arrive at a value for $\bar{v}(\varphi)$. And the function described in *this* way meets conditions 0–5 listed at the start of Section 1.2. And not only that, we know how, given φ and the values of v at its sentence symbols, to carry out the calculation of $\bar{v}(\varphi)$.

Thus, using the parsing algorithm, we can make a function $\bar{v}$ as described in Theorem 12A. And there can be only one such $\bar{v}$; cf. Exercise 14 in Section 1.2.

The only reason the existence of $\bar{v}$ is an issue at all is that in Section 1.2 it is described by *recursion*. That is, $\bar{v}(\varphi)$ is specified by making use of the function $\bar{v}$ itself, applied to smaller formulas. In the next section, we approach the matter of defining a function by recursion more generally. By treating the subject more abstractly, we can better isolate what is at stake.

Polish Notation

It is possible to avoid both ambiguity and parentheses. This can be done by a very simple device. Instead of, for example, $(\alpha \wedge \beta)$ we use $\wedge\alpha\beta$. Let the set of P-wffs be the set generated from the sentence symbols by the five operations

$$\mathcal{D}_\neg(\alpha) = \neg\alpha, \qquad \mathcal{D}_\vee(\alpha, \beta) = \vee\alpha\beta,$$
$$\mathcal{D}_\wedge(\alpha, \beta) = \wedge\alpha\beta, \qquad \mathcal{D}_\rightarrow(\alpha, \beta) = \rightarrow\alpha\beta,$$
$$\mathcal{D}_\leftrightarrow(\alpha, \beta) = \leftrightarrow\alpha\beta.$$

For example, one P-wff is

$$\to \wedge\, \mathbf{AD} \vee \neg\,\mathbf{B} \leftrightarrow \mathbf{CB}.$$

Here the need for an algorithm to analyze the structure is quite apparent. Even for the short example above, it requires some thought to see how it was built up. We will give a unique readability theorem for such expressions in Section 2.3.

This way of writing formulas (but with N, K, A, C, and E in place of $\neg$, $\wedge$, $\vee$, $\to$, and $\leftrightarrow$, respectively) was introduced by the Polish logician Łukasiewicz. The notation is well suited to automatic processing. Computer compiler programs often begin by converting the formulas given them into Polish notation.

Omitting Parentheses

Hereafter when naming wffs, we will not feel compelled to mention explicitly every parenthesis. To establish a more compact notation, we now adopt the following conventions:

1. The outermost parentheses need not be explicitly mentioned. For example, when we write "$\mathbf{A} \wedge \mathbf{B}$" we are referring to $(\mathbf{A} \wedge \mathbf{B})$.

2. The negation symbol applies to as little as possible. For example, $\neg\,\mathbf{A} \wedge \mathbf{B}$ is $(\neg\,\mathbf{A}) \wedge \mathbf{B}$, i.e., $((\neg\,\mathbf{A}) \wedge \mathbf{B})$. It is not the same as $(\neg\,(\mathbf{A} \wedge \mathbf{B}))$.

3. The conjunction and disjunction symbols apply to as little as possible, given that convention 2 is to be observed. For example,

$$\mathbf{A} \wedge \mathbf{B} \to \neg\,\mathbf{C} \vee \mathbf{D} \quad \text{is} \quad ((\mathbf{A} \wedge \mathbf{B}) \to ((\neg\,\mathbf{C}) \vee \mathbf{D})).$$

4. Where one connective symbol is used repeatedly, grouping is to the right:

$$\alpha \wedge \beta \wedge \gamma \quad \text{is} \quad \alpha \wedge (\beta \wedge \gamma),$$

$$\alpha \to \beta \to \gamma \quad \text{is} \quad \alpha \to (\beta \to \gamma).$$

It must be admitted that these conventions violate what was said earlier about naming expressions. We can get away with this only because we no longer have any interest in naming expressions that are not wffs.

Exercises

1. Rewrite the tautologies in the "selected list" at the end of Section 1.2, but using the conventions of the present section to minimize the number of parentheses.

2. Give an example of wffs α and β and expressions γ and δ such that $(\alpha \wedge \beta) = (\gamma \wedge \delta)$ but $\alpha \neq \gamma$.

3. Carry out the argument for Lemma 13B for the case of the operation $\mathcal{E}_{\neg}$.

4. Suppose that we modify our definition of wff by omitting all *right* parentheses. Thus instead of

$$((A \wedge (\neg B)) \rightarrow (C \vee D))$$

   we use

$$((A \wedge (\neg B \rightarrow (C \vee D.$$

   Show that we still have unique readability (i.e., each wff still has only one possible decomposition). *Suggestion*: These expressions have the same number of parentheses as connective symbols.

5. The English language has a tendency to use two-part connectives: "both ... and ..." "either ... or ..." "if ..., then ...." How does this affect unique readability in English?

6. We have given an algorithm for analyzing a wff by constructing its tree from the top down. There are also ways of constructing the tree from the bottom up. This can be done by looking through the formula for innermost pairs of parentheses. Give a complete description of an algorithm of this sort.

7. Suppose that left and right parentheses are indistinguishable. Thus, instead of $(\alpha \vee (\beta \wedge \gamma))$ we have $|\alpha \vee |\beta \wedge \gamma||$. Do formulas still have unique decomposition?

SECTION 1.4
Induction and Recursion[1]

Induction

There is one special type of construction that occurs frequently both in logic and in other branches of mathematics. We may want to construct a certain subset of a set U by starting with some initial elements of U, and applying certain operations to them over and over again. The set we seek will be the smallest set containing the initial elements and closed under the operations. Its members will be those elements of U which can be built up from the initial elements by applying the operations some finite number of times.

---

[1] On the one hand, the concepts in this section are important, and they arise in many places throughout mathematics. On the other hand, readers may want to postpone — not skip — study of this section.

In the special case of immediate interest to us, U is the set of expressions, the initial elements are the sentence symbols, and the operations are $\mathcal{E}_\neg$, $\mathcal{E}_\wedge$, etc. The set to be constructed is the set of wffs. But we will encounter other special cases later, and it will be helpful to view the situation abstractly here.

To simplify our discussion, we will consider an initial set $B \subseteq U$ and a class $\mathcal{F}$ of functions containing just two members f and g, where

$$f : U \times U \to U \quad \text{and} \quad g : U \to U.$$

Thus f is a binary operation on U and g is a unary operation. (Actually $\mathcal{F}$ need not be finite; it will be seen that our simplified discussion here is, in fact, applicable to a more general situation. $\mathcal{F}$ can be any set of relations on U, and in Chapter 2 this greater generality will be utilized. But the case discussed here is easier to visualize and is general enough to illustrate the ideas. For a less restricted version, see Exercise 3.)

If B contains points a and b, then the set C we wish to construct will contain, for example,

$$b, \quad f(b, b), \quad g(a), \quad f(g(a), f(b, b)), \quad g(f(g(a), f(b, b))).$$

Of course these might not all be distinct. The idea is that we are given certain bricks to work with, and certain types of mortar, and we want C to contain just the things we are able to build.

In defining C more formally, we have our choice of two definitions. We can define it "from the top down" as follows: Say that a subset S of U is *closed* under f and g iff whenever elements x and y belong to S, then so also do $f(x, y)$ and $g(x)$. Say that S is *inductive* iff $B \subseteq S$ and S is closed under f and g. Let $C^\star$ be the intersection of all the inductive subsets of U; thus $x \in C^\star$ iff x belongs to every inductive subset of U. It is not hard to see (and the reader should check) that $C^\star$ is itself inductive. Furthermore, $C^\star$ is the smallest such set, being included in all the other inductive sets.

The second (and equivalent) definition works "from the bottom up." We want $C_\star$ to contain the things that can be reached from B by applying f and g a finite number of times. Temporarily define a *construction sequence* to be a finite sequence $\langle x_1, \ldots, x_n \rangle$ of elements of U such that for each $i \leq n$ we have at least one of

$$
\begin{aligned}
&x_i \in B, \\
&x_i = f(x_j, x_k) \quad \text{for some} \quad j < i, k < i, \\
&x_i = g(x_j) \qquad\quad \text{for some} \quad j < i.
\end{aligned}
$$

In other words, each member of the sequence either is in B or results from *earlier* members by applying f or g. Then let $C_\star$ be the set of all points x such that some construction sequence ends with x.

Let C_n be the set of points x such that some construction sequence of length n ends with x. Then $C_1 = B$,

$$C_1 \subseteq C_2 \subseteq C_3 \subseteq \cdots ,$$

and $C_\star = \bigcup_n C_n$. For example, $g(f(a, f(b, b)))$ is in C_5 and hence in $C_\star$, as can be seen by contemplating the tree shown:

$$g(f(a, f(b, b)))$$
$$|$$
$$f(a, f(b, b))$$
$$\diagup \quad \diagdown$$
$$a \qquad f(b, b)$$
$$\diagup \quad \diagdown$$
$$b \qquad b$$
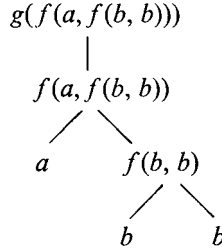
We obtain a construction sequence for $g(f(a, f(b, b)))$ by squashing this tree into a linear sequence.

EXAMPLES

1. The natural numbers. Let U be the set of all real numbers, and let $B = \{0\}$. Take one operation S, where $S(x) = x + 1$. Then

$$C_\star = \{0, 1, 2, \ldots\}.$$

The set $C_\star$ of natural numbers contains exactly those numbers obtainable from 0 by applying the successor operation repeatedly.

2. The integers. Let U be the set of all real numbers; let $B = \{0\}$. This time take two operations, the successor operation S and the predecessor operation P:

$$S(x) = x + 1 \quad \text{and} \quad P(x) = x - 1.$$

Now $C_\star$ contains all the integers,

$$C_\star = \{\ldots, -2, -1, 0, 1, 2, \ldots\}.$$

Notice that there is more than one way of obtaining 2 as a member of $C_\star$. For 2 is $S(S(0))$, but it is also $S(P(S(S(0))))$.

3. The algebraic functions. Let U contain all functions whose domain and range are each sets of real numbers. Let B contain the identity function and all constant functions. Let $\mathcal{F}$ contain the operations (on functions) of addition, multiplication, division, and root extraction. Then $C_\star$ is the class of algebraic functions.

4. The well-formed formulas. Let U be the set of all expressions and let B be the set of sentence symbols. Let $\mathcal{F}$ contain the five formula-building operations on expressions: $\mathcal{E}_\neg, \mathcal{E}_\wedge, \mathcal{E}_\vee, \mathcal{E}_\rightarrow$, and $\mathcal{E}_\leftrightarrow$. Then $C_\star$ is the set of all wffs.

At this point we should verify that our two definitions are actually equivalent, i.e., that $C^\star = C_\star$.

To show that $C^\star \subseteq C_\star$ we need only check that $C_\star$ is inductive, i.e., that $B \subseteq C_\star$ and $C_\star$ is closed under the functions. Clearly $B = C_1 \subseteq C_\star$. If x and y are in $C_\star$, then we can concatenate their construction sequences and append $f(x, y)$ to obtain a construction sequence placing $f(x, y)$ in $C_\star$. Similarly, $C_\star$ is closed under g.

Finally, to show that $C_\star \subseteq C^\star$ we consider a point in $C_\star$ and a construction sequence $\langle x_0, \ldots, x_n \rangle$ for it. By ordinary induction on i, we can see that $x_i \in C^\star$, $i \leq n$. First $x_0 \in B \subseteq C^\star$. For the inductive step we use the fact that $C^\star$ is closed under the functions. Thus we conclude that

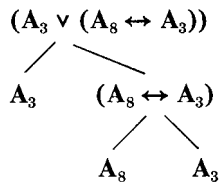$$\bigcup_n C_n = C_\star = C^\star = \bigcap \{ S \mid S \quad \text{is inductive} \}.$$

(A parenthetical remark: Suppose our present study is embedded in axiomatic set theory, where the natural numbers are usually defined from the top down. Then our definition of $C_\star$ (employing finiteness and hence natural numbers) is not really different from our definition of $C^\star$. But we are not working within axiomatic set theory; we are working within informal mathematics. And the notion of natural number seems to be a solid, well-understood intuitive concept.)

Since $C^\star = C_\star$, we call the set simply C and refer to it as the set *generated from B by* the functions in $\mathcal{F}$. We will often want to prove theorems by using the following:

> **INDUCTION PRINCIPLE**   Assume that C is the set generated from B by the functions in $\mathcal{F}$. If S is a subset of C that includes B and is closed under the functions of $\mathcal{F}$ then $S = C$.
>
> PROOF. S is inductive, so $C = C^\star \subseteq S$. We are given the other inclusion.   ⊣

The special case now of interest to us is, of course, Example 4. Here C is the class of wffs generated from the set of sentence symbols by the formula-building operations. This special case has interesting features of its own. Both α and β are proper segments of $\mathcal{E}_\wedge(\alpha, \beta)$, i.e., of $(\alpha \wedge \beta)$. More generally, if we look at the family tree of a wff, we see that each constituent is a proper segment of the end product.

$$(\mathbf{A_3} \vee (\mathbf{A_8} \leftrightarrow \mathbf{A_3}))$$

$$\mathbf{A_3} \qquad (\mathbf{A_8} \leftrightarrow \mathbf{A_3})$$

$$\mathbf{A_8} \qquad \mathbf{A_3}$$

Suppose, for example, that we temporarily call an expression *special* if the only sentence symbols in it are among $\{A_2, A_3, A_5\}$ and the only connective symbols in it are among $\{\neg, \rightarrow\}$. Then no special wff requires A_9 or $\mathcal{E}_\wedge$ for its construction. In fact, every special wff belongs to the set C_s generated from $\{A_2, A_3, A_5\}$ by $\mathcal{E}_\neg$ and $\mathcal{E}_\rightarrow$. (We can use the induction principle to show that every wff either belongs to C_s or is not special.)

Recursion

We return now to the more abstract case. There is a set U (such as the set of all expressions), a subset B of U (such as the set of sentence symbols), and two functions f and g, where

$$f : U \times U \rightarrow U \quad \text{and} \quad g : U \rightarrow U.$$

C is the set generated from B by f and g.

The problem we now want to consider is that of defining a function on C recursively. That is, we suppose we are given

1. Rules for computing $\bar{h}(x)$ for $x \in B$.
2a. Rules for computing $\bar{h}(f(x, y))$, making use of $\bar{h}(x)$ and $\bar{h}(y)$.
2b. Rules for computing $\bar{h}(g(x))$, making use of $\bar{h}(x)$.

(For example, this is the situation discussed in Section 1.2, where $\bar{h}$ is the extension of a truth assignment for B.) It is not hard to see that there can be at most one function $\bar{h}$ on C meeting all the given requirements.

But it is possible that no such $\bar{h}$ exists; the rules may be contradictory. For example, let

$$\begin{aligned} U &= \text{the set of real numbers,} \\ B &= \{0\}, \\ f(x, y) &= x \cdot y, \\ g(x) &= x + 1. \end{aligned}$$

Then C is the set of natural numbers. Suppose we impose the following requirements on $\bar{h}$:

1. $\bar{h}(0) = 0$.
2a. $\bar{h}(f(x, y)) = f(\bar{h}(x), \bar{h}(y))$.
2b. $\bar{h}(g(x)) = \bar{h}(x) + 2$.

Then no such function $\bar{h}$ can exist. (Try computing $\bar{h}(1)$, noting that we have both $1 = g(0)$ and $1 = f(g(0), g(0))$.)

A similar situation is encountered in algebra.[2] Suppose that you have a group G that is generated from B by the group multiplication

---

[2] It is hoped that examples such as this will be useful to the reader with some algebraic experience. The other readers will be glad to know that these examples are merely illustrative and not essential to our development of the subject.

and inverse operation. Then an arbitrary map of B into a group H is not necessarily extendible to a homomorphism of the entire group G into H. But if G happens to be a free group with set B of independent generators, then any such map is extendible to a homomorphism of the entire group.

Say that C is *freely* generated from B by f and g iff in addition to the requirements for being generated, the restrictions f_C and g_C of f and g to C meet the following conditions:

1. f_C and g_C are one-to-one.
2. The range of f_C, the range of g_C, and the set B are pairwise disjoint.

The main result of this section, the recursion theorem, says that if C is freely generated then a function h on B always has an extension $\bar{h}$ on C that follows the sorts of rules considered above.

RECURSION THEOREM   Assume that the subset C of U is freely generated from B by f and g, where

$$f : U \times U \rightarrow U,$$
$$g : U \rightarrow U.$$

Further assume that V is a set and F, G, and h functions such that

$$h : B \rightarrow V,$$
$$F : V \times V \rightarrow V,$$
$$G : V \rightarrow V.$$

Then there is a unique function

$$\bar{h} : C \rightarrow V$$

such that

(i) For x in B, $\bar{h}(x) = h(x)$;
(ii) For x, y in C,

$$\bar{h}(f(x, y)) = F(\bar{h}(x), \bar{h}(y)),$$
$$\bar{h}(g(x)) = G(\bar{h}(x)).$$

Viewed algebraically, the conclusion of this theorem says that any map h of B into V can be extended to a homomorphism $\bar{h}$ from C (with operations f and g) into V (with operations F and G).

If the content of the recursion theorem is not immediately apparent, try looking at it chromatically. You want to have a function $\bar{h}$ that paints each member of C some color. You have before you

1. h, telling you how to color the initial elements in B;
2. F, which tells you how to combine the color of x and y to obtain the color of $f(x, y)$ (i.e., it gives $\bar{h}(f(x, y))$ in terms of $\bar{h}(x)$ and $\bar{h}(y)$);

3. G, which similarly tells you how to convert the color of x into the color of $g(x)$.

The danger is that of a conflict in which, for example, F is saying "green" but G is saying "red" for the same point (unlucky enough to be equal both to $f(x, y)$ and $g(z)$ for some x, y, z). But if C is *freely* generated, then this danger is avoided.

EXAMPLES.   Consider again the examples of the preceding subsection.

1. $B = \{0\}$ with one operation, the successor operation S. Then C is the set $\mathbb{N}$ of natural numbers. Since the successor operation is one-to-one and 0 is not in its range, C is freely generated from $\{0\}$ by S. Therefore, by the recursion theorem, for any set V, any $a \in V$, and any $F : V \to V$ there is a unique $\overline{h} : \mathbb{N} \to V$ such that $\overline{h}(0) = a$ and $\overline{h}(S(x)) = F(\overline{h}(x))$ for each $x \in \mathbb{N}$. For example, there is a unique $\overline{h} : \mathbb{N} \to \mathbb{N}$ such that $\overline{h}(0) = 0$ and $\overline{h}(S(x)) = 1 - \overline{h}(x)$. This function has the value 0 at even numbers and the value 1 at odd numbers.

2. The integers are generated from $\{0\}$ by the successor and predecessor operations but not freely generated.

3. Freeness fails also for the generation of the algebraic functions in the manner described.

4. The wffs are freely generated from the sentence symbols by the five formula-building operations. This fact is implicit in the parsing algorithm of the preceding section; we now want to focus on it here:

UNIQUE READABILITY THEOREM   The five formula-building operations, when restricted to the set of wffs,

(a) Have ranges that are disjoint from each other and from the set of sentence symbols, and
(b) Are one-to-one.

In other words, the set of wffs is *freely* generated from the set of sentence symbols by the five operations.

PROOF.   To show that the restriction of $\mathcal{E}_\wedge$ is one-to-one, suppose that

$$(\alpha \wedge \beta) = (\gamma \wedge \delta),$$

where α, β, γ, and δ are wffs. Delete the first symbol of each sequence, obtaining

$$\alpha \wedge \beta) = \gamma \wedge \delta).$$

Then we must have $\alpha = \gamma$, lest one be a proper initial segment of the other (in contradiction to Lemma 13B). And then it follows at

once that $\beta = \delta$. The same argument applies to $\mathcal{E}_\vee$, $\mathcal{E}_\rightarrow$, and $\mathcal{E}_\leftrightarrow$; for $\mathcal{E}_\neg$ a simpler argument suffices.

A similar line of reasoning tells us that the operations have disjoint ranges. For example, if

$$(\alpha \wedge \beta) = (\gamma \rightarrow \delta)$$

where α, β, γ, and δ are wffs, then as in the above paragraph we have $\alpha = \gamma$. But that implies that $\wedge = \rightarrow$, contradicting the fact that our symbols are distinct. Hence $\mathcal{E}_\wedge$ and $\mathcal{E}_\rightarrow$ (when restricted to wffs) have disjoint ranges. Similarly for any two binary connectives.

The remaining cases are simple. If $(\neg \alpha) = (\beta \wedge \gamma)$, then β begins with $\neg$, which no wff does. No sentence symbol is a sequence of symbols beginning with (. $\dashv$

Now let us return to the question of extending a truth assignment v to $\bar{v}$. First consider the special case where v is a truth assignment for the set of all sentence symbols. Then by applying the unique readability theorem and the recursion theorem we conclude that there is a unique extension $\bar{v}$ to the set of all wffs with the desired properties.

Next take the general case where v is a truth assignment for a set S of sentence symbols. The set $\bar{S}$ generated from S by the five formula-building operations is freely generated, as a consequence of the unique readability theorem. So by the recursion theorem there is a unique extension $\bar{v}$ of v to that set, having the desired properties.

EXAMPLE.   We can apply the recursion theorem to establish that there is a unique function h defined on the set of wffs such that

$$\begin{aligned}
\bar{h}(A) &= 1 \quad \text{for a sentence symbol} \quad A, \\
\bar{h}((\neg \alpha)) &= 3 + \bar{h}(\alpha), \\
\bar{h}((\alpha \wedge \beta)) &= 3 + \bar{h}(\alpha) + \bar{h}(\beta),
\end{aligned}$$

and similarly for $\vee$, $\rightarrow$, and $\leftrightarrow$. This function gives the length of each wff.

PROOF OF THE RECURSION THEOREM.   The idea is to let $\bar{h}$ be the union of many approximating functions. Temporarily call a function v (which maps part of C into V) *acceptable* if it meets the conditions imposed on $\bar{h}$ by (i) and (ii). More precisely, v is acceptable iff the domain of v is a subset of C, the range a subset of V, and for any x and y in C:

(i′) If x belongs to B and to the domain of v, then $v(x) = h(x)$.
(ii′) If $f(x, y)$ belongs to the domain of v, then so do x and y, and $v(f(x, y)) = F(v(x), v(y))$. If $g(x)$ belongs to the domain of v, then so does x, and $v(g(x)) = G(v(x))$.

Let K be the collection of all acceptable functions, and let $\overline{h} = \bigcup K$, the union of all the acceptable functions. Thus

$$\begin{aligned} \langle x, z \rangle \in \overline{h} \quad &\text{iff} \quad \langle x, z \rangle \text{ belongs to some acceptable } v \\ &\text{iff} \quad v(x) = z \text{ for some acceptable } v. \end{aligned} \qquad (1.1)$$

We claim that $\overline{h}$ meets our requirements. The argument is set-theoretic, and comprises four steps. First, here is an outline of four steps:

1. We claim that $\overline{h}$ is a function (i.e., that it is single-valued). Let

$$\begin{aligned} S = &\{ x \in C \mid \text{ for at most one } z, \ \langle x, z \rangle \in \overline{h} \} \\ = &\{ x \in C \mid \text{ all acceptable functions defined at } x \text{ agree there} \} \end{aligned}$$

It is easy to verify that S is inductive, by using (i′) and (ii′). Hence $S = C$ and $\overline{h}$ is a function.

2. We claim that $\overline{h} \in K$; i.e., that $\overline{h}$ itself is an acceptable function. This follows fairly easily from the definition of $\overline{h}$ and the fact that it is a function.

3. We claim that $\overline{h}$ is defined throughout C. It suffices to show that the domain of $\overline{h}$ is inductive. It is here that the assumption of freeness is used. For example, one case is the following: Suppose that x is in the domain of $\overline{h}$. Then $\overline{h}; \langle g(x), G(\overline{h}(x)) \rangle$ is acceptable. (The freeness is required in showing that it is acceptable.) Consequently, $g(x)$ is in the domain of $\overline{h}$.

4. We claim that $\overline{h}$ is unique. For given two such functions, let S be the set on which they agree. Then S is inductive, and so equals C.   ⊣

Now for the details.

1. As above, let

$$\begin{aligned} S = &\{ x \in C \mid \text{ for at most one } z, \ \langle x, z \rangle \in \overline{h} \} \\ = &\{ x \in C \mid \text{ all acceptable functions defined at } x \text{ agree there} \} \end{aligned}$$

Toward showing that S is inductive, first consider some x in B. Suppose that v_1 and v_2 are acceptable functions defined at x; we seek to show that $v_1(x) = v_2(x)$. But condition (i′) tells us that both $v_1(x)$ and $v_2(x)$ must equal $h(x)$, so indeed $v_1(x) = v_2(x)$. This shows that $x \in S$; since x was an arbitrary member of B we have $B \subseteq S$.

Secondly we must check that S is closed under f and g. So suppose that some x and y are in S; we ask whether $f(x, y)$ is in S. So suppose that v_1 and v_2 are acceptable functions defined at

$f(x, y)$; we seek to show that they agree there. But condition (ii′) tells us that $v_1(f(x, y)) = F(v_1(x), v_1(y))$ and $v_2(f(x, y)) = F(v_2(x), v_2(y))$. And because x and y are in S, we have $v_1(x) = v_2(x)$ and $v_1(y) = v_2(y)$ (and these are defined). So we conclude that $v_1(f(x, y)) = v_2(f(x, y))$. This shows that $f(x, y) \in S$. Hence S is closed under f. A similar argument shows that S is closed under g.

Thus S is inductive and so $S = C$. This shows that $\overline{h}$ is single-valued, i.e., is a function. Because $\overline{h}$ includes every acceptable function as a subset, we can say that

$$\overline{h}(x) = v(x)$$

whenever v is an acceptable function and $x \in \operatorname{dom} v$.

2. We claim that $\overline{h}$ is acceptable. Clearly $\operatorname{dom} \overline{h} \subseteq C$ and $\operatorname{ran} \overline{h} \subseteq V$ (by (∗)), and we have just verified that $\overline{h}$ is a function. It remains to check that $\overline{h}$ satisfies conditions (i′) and (ii′).

First we examine (i′). Assume $x \in B$ and $x \in \operatorname{dom} \overline{h}$ (so that $\langle x, \overline{h}(x) \rangle \in \overline{h}$). There must be some acceptable v with $v(x) = \overline{h}(x)$. Because v satisfies (i′), we have $v(x) = h(x)$ whence $\overline{h}(x) = h(x)$. So $\overline{h}$ satisfies (i′).

Secondly we examine (ii′). Assume that $f(x, y) \in \operatorname{dom} \overline{h}$. Again there must be some acceptable v with $v(f(x, y)) = \overline{h}(f(x, y))$. Because v satisfies (ii′), we have $v(f(x, y)) = F(v(x), v(y))$. Now $\overline{h}(x) = v(x)$ and $\overline{h}(y) = v(y)$ and hence

$$\overline{h}(f(x, y)) = v(f(x, y)) = F(v(x), v(y)) = F(\overline{h}(x), \overline{h}(y)).$$

In a similar way, we find that $\overline{h}(g(x)) = G(\overline{h}(x))$ whenever $g(x) \in \operatorname{dom} \overline{h}$. Hence $\overline{h}$ meets condition (ii′) and so is acceptable.

3. Next we must show that $\operatorname{dom} \overline{h}$ is inductive. First consider a point x in B. Then the set $\{\langle x, h(x) \rangle\}$ is a (small) acceptable function. For it clearly satisfies (i′). It also satisfies (ii′) because $x \notin \operatorname{ran} f_C$ and $x \notin \operatorname{ran} g_C$. Thus $\{\langle x, h(x) \rangle\}$ is acceptable and therefore is included in $\overline{h}$. Hence $x \in \operatorname{dom} \overline{h}$. This shows that $B \subseteq \operatorname{dom} \overline{h}$.

We further claim that $\operatorname{dom} \overline{h}$ is closed under f and g. Toward this end, consider any s and t in $\operatorname{dom} \overline{h}$. We hope that $f(s, t) \in \operatorname{dom} \overline{h}$. But if not, then let

$$v = \overline{h} \cup \{\langle f(s, t), F(\overline{h}(s), \overline{h}(t)) \rangle\},$$

the result of adding this one additional pair to $\overline{h}$. It is clear that v is a function, $\operatorname{dom} v \subseteq C$, and $\operatorname{ran} v \subseteq V$. We claim that v satisfies (i′) and (ii′).

First take (i′). If $x \in B \cap \operatorname{dom} v$ then $x \neq f(s, t)$, by freeness. Hence $x \in \operatorname{dom} \overline{h}$ and we have $v(x) = \overline{h}(x) = h(x)$.

Next take (ii′). Assume that $f(x, y) \in \operatorname{dom} v$ for some x and y in C. If $f(x, y) \in \operatorname{dom} \overline{h}$ then $v(f(x, y)) = \overline{h}(f(x, y)) = F(\overline{h}(x), \overline{h}(y)) = F(v(x), v(y))$ since $\overline{h}$ is acceptable. The other possibility is that $f(x, y) = f(s, t)$. Then by freeness we have $x = s$ and $y = t$, and we know that these points are in $\operatorname{dom} \overline{h} \subseteq \operatorname{dom} v$. By construction,

$$
\begin{aligned}
v(f(s, t)) &= F(\overline{h}(s), \overline{h}(t)) \\
&= F(v(s), v(t)).
\end{aligned}
$$

Finally suppose that $g(x) \in \operatorname{dom} v$ for x in C. Then by freeness we have $g(x) \neq f(s, t)$. Hence $g(x) \in \operatorname{dom} \overline{h}$, and consequently $v(g(x)) = \overline{h}(g(x)) = G(\overline{h}(x)) = G(v(x))$.

Thus v is acceptable. But that tells us that $v \subseteq \overline{h}$, so that $f(s, t) \in \operatorname{dom} \overline{h}$ after all.

A similar argument shows that $\operatorname{dom} \overline{h}$ is closed under g as well. Hence $\operatorname{dom} \overline{h}$ is inductive and therefore coincides with C.

4. To show that $\overline{h}$ is unique, suppose that $\overline{h}_1$ and $\overline{h}_2$ both satisfy the conclusion of the theorem. Let S be the set on which they agree:

$$
S = \{x \in C \mid \overline{h}_1(x) = \overline{h}_2(x)\}.
$$

Then it is not hard to verify that S is inductive. Consequently $S = C$ and $\overline{h}_1 = \overline{h}_2$.   ⊣

One final comment on induction and recursion: The induction principle we have stated is not the only one possible. It is entirely possible to give proofs by induction (and definitions by recursion) on the length of expressions, the number of places at which connective symbols occur, etc. Such methods are inherently less basic but may be necessary in some situations.

Exercises

1. Suppose that C is generated from a set $B = \{a, b\}$ by the binary operation f and unary operation g. List all the members of C_2. How many members might C_3 have? C_4?

2. Obviously $(\mathbf{A}_3 \rightarrow \wedge \mathbf{A}_4)$ is not a wff. But prove that it is not a wff.

3. We can generalize the discussion in this section by requiring of $\mathcal{F}$ only that it be a class of relations on U. $C_\star$ is defined as before, except that $\langle x_0, x_1, \ldots, x_n \rangle$ is now a construction sequence provided that for each $i \leq n$ we have either $x_i \in B$ or $\langle x_{j_1}, \ldots, x_{j_k}, x_i \rangle \in R$ for some $R \in \mathcal{F}$ and some $j_1, \ldots, j_k$ all less than i. Give the correct definition of $C^\star$ and show that $C^\star = C_\star$.

SECTION 1.5
Sentential Connectives

We have thus far employed five sentential connective symbols. Even in the absence of a general definition of "connective," it is clear that the five familiar ones are not the only ones possible. Would we gain anything by adding more connectives to the language? Would we lose anything by omitting some we already have?

In this section we make these questions precise and give some answers. First consider an informal example. We could expand the language by adding a three-place sentential connective symbol #, called the majority symbol. We allow now as a wff the expression $(\#\alpha\beta\gamma)$ whenever α, β, and γ are wffs. In other words, we add a sixth formula-building operator to our list:

$$\mathcal{E}_\#(\alpha, \beta, \gamma) = (\#\alpha\beta\gamma).$$

Then we must give the interpretation of this symbol. That is, we must say how to compute $\overline{v}((\#\alpha\beta\gamma))$, given the values $\overline{v}(\alpha)$, $\overline{v}(\beta)$, and $\overline{v}(\gamma)$. We choose to define

$\overline{v}((\#\alpha\beta\gamma))$ is to agree with the majority of $\overline{v}(\alpha), \overline{v}(\beta), \overline{v}(\gamma).$

We claim that this extension has gained us nothing, in the following precise sense: For any wff in the extended language, there is a tautologically equivalent wff in the original language. (On the other hand, the wff in the original language may be much longer than the wff in the extended language.) We will prove this (in a more general situation) below; here we just note that it relies on the fact that $(\#\alpha\beta\gamma)$ is tautologically equivalent to

$$(\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \vee (\beta \wedge \gamma).$$

(We note parenthetically that our insistence that $\overline{v}((\#\alpha\beta\gamma))$ be calculable from $\langle \overline{v}(\alpha), \overline{v}(\beta), \overline{v}(\gamma) \rangle$ plays a definite role here. In everyday speech, there are unary operators like "it is possible that" or "I believe that." We can apply one of these operators to a sentence, producing a new sentence whose truth or falsity cannot be determined solely on the basis of the truth or falsity of the original one.)

In generalizing the foregoing example, the formal language will be more of a hindrance than a help. We can restate everything using only functions. Say that a k-place *Boolean function* is a function from $\{F, T\}^k$ into $\{F, T\}$. (A *Boolean function* is then anything which is a k-place Boolean function for some k. We stretch this slightly by permitting F and T themselves to be 0-place Boolean functions.) Some sample

Boolean functions are defined by the equations (where $X \in \{F, T\}$)

$$I_i^n(X_1, \ldots, X_n) = X_i,$$
$$N(F) = T, \ N(T) = F,$$
$$K(T, T) = T, \quad K(F, X) = K(X, F) = F,$$
$$A(F, F) = F, \quad A(T, X) = A(X, T) = T,$$
$$C(T, F) = F, \quad C(F, X) = C(X, T) = T,$$
$$E(X, X) = T, \quad E(T, F) = E(F, T) = F.$$

From a wff α we can extract a Boolean function. For example, if α is the wff $\mathbf{A}_1 \wedge \mathbf{A}_2$, then we can make a table, Table V. The 2^2 lines of the table correspond to the 2^2 truth assignments for $\{\mathbf{A}_1, \mathbf{A}_2\}$. For each of the 2^2 pairs $\vec{X}$, we set $B_\alpha(\vec{X})$ equal to the truth value α receives when its sentence symbols are given the values indicated by $\vec{X}$.

TABLE V

$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_1 \wedge \mathbf{A}_2$	
F	F	F	$B_\alpha(F, F) = F$
F	T	F	$B_\alpha(F, T) = F$
T	F	F	$B_\alpha(T, F) = F$
T	T	T	$B_\alpha(T, T) = T$

In general, suppose that α is a wff whose sentence symbols are at most $\mathbf{A}_1, \ldots, \mathbf{A}_n$. We define an n-place Boolean function B_α^n (or just B_α if n seems unnecessary), the Boolean function *realized* by α, by

$B_\alpha^n(X_1, \ldots, X_n) =$ the truth value given to α when
$\qquad\qquad \mathbf{A}_1, \ldots, \mathbf{A}_n \quad$ are given the values $\quad X_1, \ldots, X_n$.

Or, in other words, $B_\alpha^n(X_1, \ldots, X_n) = \bar{v}(\alpha)$, where v is the truth assignment for $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\}$ for which $v(\mathbf{A}_i) = X_i$. Thus B_α^n comes from looking at $\bar{v}(\alpha)$ as a function of v, with α fixed.

For example, the Boolean functions listed previously are obtainable in this way:

$$I_i^n = B_{\mathbf{A}_i}^n$$
$$N = B_{\neg \mathbf{A}_1}^1,$$
$$K = B_{\mathbf{A}_1 \wedge \mathbf{A}_2}^2,$$
$$A = B_{\mathbf{A}_1 \vee \mathbf{A}_2}^2,$$
$$C = B_{\mathbf{A}_1 \rightarrow \mathbf{A}_2}^2,$$
$$E = B_{\mathbf{A}_1 \leftrightarrow \mathbf{A}_2}^2.$$

From these functions we can compose others. For example,

$$B_{\neg \mathbf{A}_1 \vee \neg \mathbf{A}_2}^2(X_1, X_2) = A(N(I_1^2(X_1, X_2)), N(I_2^2(X_1, X_2))).$$

(The right-hand side of this equation can be compared with the result of putting $\neg A_1 \vee \neg A_2$ into Polish notation.) We will shortly come to the question whether every Boolean function is obtainable in this fashion.

As the theorem below states, in shifting attention from wffs to the Boolean functions they realize, we have in effect identified tautologically equivalent wffs. Impose an ordering on $\{F, T\}$ by defining $F < T$. (If $F = 0$ and $T = 1$, then this is the natural order.)

THEOREM 15A Let α and β be wffs whose sentence symbols are among $A_1, \ldots, A_n$. Then

(a) $\alpha \models \beta$ iff for all $\vec{X} \in \{F, T\}^n$, $B_\alpha(\vec{X}) \leq B_\beta(\vec{X})$.
(b) $\alpha \models\!\models \beta$ iff $B_\alpha = B_\beta$.
(c) $\models \alpha$ iff B_α is the constant function with value T.

PROOF OF (A). $\alpha \models \beta$ iff for all 2^n truth assignments v for $A_1, \ldots, A_n$, whenever $\overline{v}(\alpha) = T$, then also $\overline{v}(\beta) = T$. (This is true even if the sentence symbols in α and β do not include all of $A_1, \ldots, A_n$; cf. Exercise 6 of Section 1.2.) Thus

$$\alpha \models \beta \quad \text{iff for all } 2^n \text{ assignments } v, \quad \overline{v}(\alpha) = T \Rightarrow \overline{v}(\beta) = T,$$
$$\text{iff for all } 2^n \text{ } n\text{-tuples } \vec{X}, \quad B_\alpha^n(\vec{X}) = T \Rightarrow B_\beta^n(\vec{X}) = T,$$
$$\text{iff for all } 2^n \text{ } n\text{-tuples } \vec{X}, \quad B_\alpha^n(\vec{X}) \leq B_\beta^n(\vec{X}),$$

where $F < T$. ⊣

In addition to identifying tautologically equivalent wffs, we have freed ourselves from the formal language. We are now at liberty to consider any Boolean function, whether it is realized by a wff or not. But this freedom is only apparent:

THEOREM 15B Let G be an n-place Boolean function, $n \geq 1$. We can find a wff α such that $G = B_\alpha^n$, i.e., such that α realizes the function G.

The primary reason for introducing Boolean functions is to allow us to formulate this theorem. The theorem was stated by Emil Post in 1921.

PROOF. Case I: G is the constant function with value F. Let $\alpha = A_1 \wedge \neg A_1$.

Case II: Otherwise there are k points at which G has the value T, $k > 0$. List these:

$$\vec{X}_1 = \langle X_{11}, X_{12}, \ldots, X_{1n} \rangle,$$
$$\vec{X}_2 = \langle X_{21}, X_{22}, \ldots, X_{2n} \rangle,$$
$$\ldots$$
$$\vec{X}_k = \langle X_{k1}, X_{k2}, \ldots, X_{kn} \rangle.$$

Let

$$\beta_{ij} = \begin{cases} \mathbf{A}_j & \text{iff} \quad X_{ij} = T, \\ (\neg \mathbf{A}_j) & \text{iff} \quad X_{ij} = F, \end{cases}$$
$$\gamma_i = \beta_{i1} \wedge \cdots \wedge \beta_{in},$$
$$\alpha = \gamma_i \vee \gamma_2 \vee \cdots \vee \gamma_k.$$

We claim that $G = B_\alpha^n$.

At this point it might be helpful to consider a concrete example. Let G be the three-place Boolean function as follows:

$$G(F, F, F) = F,$$
$$G(F, F, T) = T,$$
$$G(F, T, F) = T,$$
$$G(F, T, T) = F,$$
$$G(T, F, F) = T,$$
$$G(T, F, T) = F,$$
$$G(T, T, F) = F,$$
$$G(T, T, T) = T.$$

Then the list of triples at which G assumes the value T has four members:

$$FFT \quad \neg \mathbf{A}_1 \wedge \neg \mathbf{A}_2 \wedge \mathbf{A}_3,$$
$$FTF \quad \neg \mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \neg \mathbf{A}_3,$$
$$TFF \quad \mathbf{A}_1 \wedge \neg \mathbf{A}_2 \wedge \neg \mathbf{A}_3,$$
$$TTT \quad \mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \mathbf{A}_3.$$

To the right of each triple above is written the corresponding conjunction γ_i. Then α is the formula

$$(\neg \mathbf{A}_1 \wedge \neg \mathbf{A}_2 \wedge \mathbf{A}_3) \vee (\neg \mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \neg \mathbf{A}_3) \vee$$
$$(\mathbf{A}_1 \wedge \neg \mathbf{A}_2 \wedge \neg \mathbf{A}_3) \vee (\mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \mathbf{A}_3).$$

Notice how α lists explicitly the triples at which G assumes the value T.

To return to the proof of the theorem, note first that $B_\alpha^n(\vec{X}_i) = T$ for $1 \leq i \leq k$. (For the truth assignment corresponding to $\vec{X}_i$ satisfies γ_i and hence satisfies α.) On the other hand, only one truth assignment for $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\}$ can satisfy γ_i, whence only k such truth assignments can satisfy α. Hence $B_\alpha^n(\vec{X}) = F$ for the $2^n - k$ other n-tuples $\vec{Y}$. Thus in all cases, $B_\alpha^n(\vec{Y}) = G(\vec{Y})$. $\dashv$

From this theorem we know that every Boolean function is realizable. Of course the α that realizes G is not unique; any tautologically equivalent wff will also realize the same function. It is sometimes of interest to choose α to be as short as possible. (In the example done above, the wff

$$\mathbf{A}_1 \leftrightarrow \mathbf{A}_2 \leftrightarrow \mathbf{A}_3$$

also realizes G.)

As a corollary to the above theorem, we may conclude that we have enough (in fact, more than enough) sentential connectives. For suppose that we expand the language by adding some exotic new sentential connectives (such as the majority connective discussed at the beginning of this section). Any wff φ of this expanded language realizes a Boolean function B_φ^n. By the above theorem we have a wff α of the original language such that $B_\varphi^n = B_\alpha^n$. Hence φ and α are tautologically equivalent, by Theorem 15A.

In fact, the proof shows that α can be of a rather special form. For one thing, the only sentential connective symbols in α are $\wedge$, $\vee$, and $\neg$. Furthermore, α is in so-called *disjunctive normal form* (abbreviated DNF). That is, α is a disjunction

$$\alpha = \gamma_1 \vee \cdots \vee \gamma_k,$$

where each γ_i is a conjunction

$$\gamma_i = \beta_{i1} \wedge \cdots \wedge \beta_{in_i}$$

and each β_{ij} is a sentence symbol or the negation of a sentence symbol. (The advantages of wffs in disjunctive normal form stem from the fact that they explicitly list the truth assignments satisfying the formula.) Thus we have the consequence:

COROLLARY 15C   For any wff φ, we can find a tautologically equivalent wff α in disjunctive normal form.

Because every function $G : \{F, T\}^n \to \{F, T\}$ for $n \geq 1$ can be realized by a wff using only the connective symbols in $\{\wedge, \vee, \neg\}$, we say that the set $\{\wedge, \vee, \neg\}$ is *complete*. (Actually the completeness is more a property of the Boolean functions K, A, and N that correspond to these symbols. But the above terminology is convenient.) Once we have a complete set of connectives, we know that any wff is tautologically equivalent to one all of whose connectives are in that set. (For given any wff φ, we can make α using those connectives and realizing B_φ. Then $\alpha \models\models \varphi$.) The completeness of $\{\wedge, \vee, \neg\}$ can be improved upon:

THEOREM 15D   Both $\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are complete.

PROOF.   We must show that any Boolean function G can be realized by a wff using only, in the first case, $\{\neg, \wedge\}$. We begin with a wff α using $\{\neg, \wedge, \vee\}$ that realizes G. It suffices to find a tautologically equivalent α' that uses only $\{\neg, \wedge\}$. For this we use De Morgan's law:

$$\beta \vee \gamma \models\models \neg(\neg\beta \wedge \neg\gamma).$$

By applying this repeatedly, we can completely eliminate $\vee$ from α.

(More formally, we can prove by induction on α that there is a tautologically equivalent α' in which only the connectives $\wedge$, $\neg$ occur. Two cases in the inductive step are

Case $\neg$: If α is $(\neg \beta)$, then let α' be $(\neg \beta')$.
Case $\vee$: If α is $(\beta \vee \gamma)$, then let α' be $\neg(\neg \beta' \wedge \neg \gamma')$. Since β' and γ' are tautologically equivalent to β and γ, respectively,

$$
\begin{aligned}
\alpha' &= \neg(\neg \beta' \wedge \neg \gamma') \\
&\models\!\!=\!\!\models \neg(\neg \beta \wedge \neg \gamma) \\
&\models\!\!=\!\!\models \beta \vee \gamma \\
&= \alpha.
\end{aligned}
$$

In future proofs that a set of connectives is complete, this induction will be omitted. Instead we will just give, for example, the method of simulating $\vee$ by using $\neg$ and $\wedge$. $\dashv$

Showing that a certain set of connectives is *not* complete is usually more difficult than showing that one is complete. The basic method is first to show (usually by induction) that for any wff α using only those connectives, the function B_α^n has some peculiarity, and secondly to show that some Boolean function lacks that peculiarity.

EXAMPLE. $\{\wedge, \rightarrow\}$ is not complete.

PROOF.   The idea is that with these connectives, if the sentence symbols are assigned T, then the entire formula is assigned T. In particular, there is nothing tautologically equivalent to $\neg \mathbf{A}$.
    In more detail, we can show by induction that for any wff α using only these connectives and having $\mathbf{A}$ as its only sentence symbol, we have $\mathbf{A} \models \alpha$. (In terms of functions, this says that $B_\alpha^1(T) = T$.) The same argument shows that $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$ is not complete. $\dashv$

For each n there are 2^{2^n} n-place Boolean functions. Hence if we identify a connective with its Boolean function (e.g., $\wedge$ with the function K mentioned before), we have 2^{2^n} n-ary connectives. We will now catalog these for $n \leq 2$.

0-ary Connectives

There are two 0-place Boolean functions, F and T. For the corresponding connective symbols we take $\bot$ and $\top$. Now an n-ary connective symbol combines with n wffs to produce a new wff. When $n = 0$, we have that $\bot$ is a wff all by itself. It differs from the sentence symbols in that $\bar{v}(\bot) = F$ for every v; i.e., $\bot$ is a logical symbol always assigned the value F. Similarly, $\top$ is a wff, and $\bar{v}(\top) = T$ for every v. Then, for example, $\mathbf{A} \rightarrow \bot$ is a wff, tautologically equivalent to $\neg \mathbf{A}$, as can be seen from a two-line truth table.

Unary Connectives

There are four unary connectives but only one of any interest. The interesting case is negation. The other three one-place Boolean functions are the identity function and the two constant functions.

Binary Connectives

There are 16 binary connectives, but only the last 10 listed in Table VI are "really binary."

TABLE VI

Symbol	Equivalent	Remarks
	$\top$	two-place constant, essentially 0-ary
	$\bot$	two-place constant, essentially 0-ary
	A	projection, essentially unary
	B	projection, essentially unary
	$\neg\,$**A**	negation, essentially unary
	$\neg\,$**B**	negation, essentially unary
$\wedge$	**A** $\wedge$ **B**	and; if $F = 0$ and $T = 1$, then this gives multiplication in the field $\{0, 1\}$
$\vee$	**A** $\vee$ **B**	or (inclusive)
$\rightarrow$	**A** $\rightarrow$ **B**	conditional
$\leftrightarrow$	**A** $\leftrightarrow$ **B**	biconditional
$\leftarrow$	**A** $\leftarrow$ **B**	reversed conditional
$+$	(**A** $\vee$ **B**) $\wedge \neg$ (**A** $\wedge$ **B**)	exclusive or, "**A** or **B** and not both"; if $F = 0$ and $T = 1$, then this gives the usual addition (modulo 2) in the field $\{0, 1\}$
$\downarrow$	$\neg$ (**A** $\vee$ **B**)	nor, "neither **A** nor **B**"
$\mid$	$\neg$ (**A** $\wedge$ **B**)	nand, "not both **A** and **B**"; the symbol is called the Sheffer stroke
$<$	$(\neg$**A**$) \wedge$ **B**	the usual ordering, where $F < T$
$>$	**A** $\wedge (\neg$**B**$)$	the usual ordering, where $F < T$

Ternary Connectives

There are 256 ternary connectives; 2 are essentially 0-ary, 6 $(= 2 \cdot \binom{3}{1})$ are essentially unary, and 30 $(= 10 \cdot \binom{3}{2})$ are essentially binary. This leaves 218 that are really ternary. We have thus far mentioned only the majority connective #. There is, similarly, the minority connective. In Exercise 7 we encounter $+^3$, ternary addition modulo 2. $+^3\alpha\beta\gamma$ is assigned the value T iff an odd number of α, β, and γ are assigned T. This formula is equivalent both to $\alpha + \beta + \gamma$ and to $\alpha \leftrightarrow \beta \leftrightarrow \gamma$. Another ternary connective arises in Exercise 8.

EXAMPLE.   {|} and {↓} are complete.

PROOF.   For |

$$\neg \alpha \models\models \alpha \mid \alpha$$
$$\alpha \vee \beta \models\models (\neg \alpha) \mid (\neg \beta).$$

Since {¬, ∨} is complete and ¬, ∨ can be simulated using only |, we conclude that {|} is complete.   ⊣

EXAMPLE.   {¬, →} is complete. In fact, of the 10 connectives that are really binary, eight have the property of forming, when added to ¬, a complete set. The two exceptions are + and ↔; see Exercise 5.

EXAMPLE.   {⊥, →} is complete. In fact, because with this set we can realize even the two 0-place Boolean functions, it is supercomplete.

Exercises

1. Let G be the following three-place Boolean function.

$$\begin{array}{ll}
G(F, F, F) = T, & G(T, F, F) = T, \\
G(F, F, T) = T, & G(T, F, T) = F, \\
G(F, T, F) = T, & G(T, T, F) = F, \\
G(F, T, T) = F, & G(T, T, T) = F.
\end{array}$$

   (a) Find a wff, using at most the connectives ∨, ∧, and ¬, that realizes G.
   (b) Then find such a wff in which connective symbols occur at not more than five places.

2. Show that | and ↓ are the only binary connectives that are complete by themselves.

3. Show that {¬, #} is not complete.

4. Let M be the ternary minority connective. (Thus $\overline{v}(M\alpha\beta\gamma)$ always disagrees with the majority of $\overline{v}(\alpha)$, $\overline{v}(\beta)$, and $\overline{v}(\gamma)$.) Show the following:
   (a) {M, ⊥} is complete.
   (b) {M} is not complete.

5. Show that {⊤, ⊥, ¬, ↔, +} is not complete. *Suggestion*: Show that any wff α using these connectives and the sentence symbols **A** and **B** has an even number of T's among the four possible values of $\overline{v}(\alpha)$.

      Remark: Another approach uses the algebra of the field {0, 1}. Any Boolean function realizable with these connectives has a certain linearity property.

6. Show that $\{\wedge, \leftrightarrow, +\}$ is complete but that no proper subset is complete.

7. Let $+^3$ be the ternary connective such that $+^3\alpha\beta\gamma$ is equivalent to $\alpha + \beta + \gamma$.
    (a) Show that $\{\top, \bot, \wedge, +^3\}$ is complete.
    (b) Show that no proper subset is complete.
        Remark: $+^3$ is the ternary *parity* connective; the condition for $\overline{v}(+\alpha_1\alpha_2\alpha_3) = T$ is that $\overline{v}(\alpha_i) = T$ for an *odd* number of i's. $+$ is the binary parity connective. The function G in the proof of Theorem 15B is the 3-place parity function.

8. Let $\mathbb{I}$ be the ternary connective such that $\mathbb{I}\alpha\beta\gamma$ is assigned the value T iff exactly one of the formulas α, β, γ is assigned the value T. Show that there are no binary connectives $\circ$ and $\triangle$ such that $\mathbb{I}\alpha\beta\gamma$ is equivalent to $(\alpha \circ \beta)\triangle\gamma$.

9. Say that a formula α is in *conjunctive normal form* (abbreviated CNF) iff it is a conjunction

$$\alpha = \gamma_1 \wedge \cdots \wedge \gamma_k$$

    where each γ_i is a disjunction

$$\gamma_i = \beta_{i1} \vee \cdots \vee \beta_{in}$$

    and each β_{ij} is either a sentence symbol or the negation of a sentence symbol.
    (a) Find a formula in conjunctive normal form that is tautologically equivalent to $\mathbf{A} \leftrightarrow \mathbf{B} \leftrightarrow \mathbf{C}$.
    (b) Show that for any formula, we can find a tautologically equivalent formula in conjunctive normal form.

10. Add the 0-place connectives $\top$, $\bot$ to our language. For each wff φ and sentence symbol A, let $\varphi_\top^A$ be the wff obtained from φ by replacing A by $\top$. Similarly for $\varphi_\bot^A$. Then let $\varphi_*^A = (\varphi_\top^A \vee \varphi_\bot^A)$. Prove the following:
    (a) $\varphi \models \varphi_*^A$.
    (b) If $\varphi \models \psi$ and A does not appear in ψ, then $\varphi_*^A \models \psi$.
    (c) The formula φ is satisfiable iff φ_*^A is satisfiable.
        Remarks: We can think of φ_*^A as trying to say everything φ says, but without being able to use the symbol A. Parts (a) and (b) state that φ_*^A is the strongest A-free consequence of φ. The formulas φ and φ_*^A are not tautologically equivalent in general, but they are "equally satisfiable" by part (c). The operation of forming φ_*^A from φ is (in another context) called *resolution* on A.

11. (Interpolation theorem) If $\alpha \models \beta$, then there is some γ all of whose sentence symbols occur both in α and in β and such that $\alpha \models \gamma \models \beta$. *Suggestion*: Use the preceding exercise.

      Remarks: There is an analogue of Exercise 11 that holds for first-order logic. But the proof in that case is very different, because there is no analogue of Exercise 10.

12. Is the set $\{\wedge, \top, \bot\}$ complete? Support your answer.

SECTION 1.6
Switching Circuits[1]

Consider an electrical device (traditionally a black box) having n inputs and one output (Fig. 1). Assume that to each input we apply a signal having one of two values and that the output has one of two values. The two possible values we call F and T. (We could also define the F value as 0 potential and choose the unit of potential so that the T value has potential 1.) Further assume that the device has no memory; i.e., the present output level depends only on the present inputs (and not on past history). Then the performance of the device is described by a Boolean function:

$$G(X_1, \ldots, X_n) = \text{the output level given the input signals } X_1, \ldots, X_n.$$
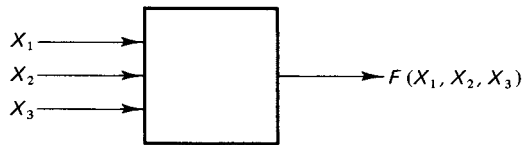

Figure 1. Electrical device with three inputs.

      Devices meeting all these assumptions constitute an essential part of digital-computer circuitry. There is, for example, the two-input AND gate, for which the output is the minimum of the inputs (where $F < T$). This device realizes the Boolean function K of the preceding section. It is convenient to attach the labels $\mathbf{A}_1$ and $\mathbf{A}_2$ to the inputs and to label the output $\mathbf{A}_1 \wedge \mathbf{A}_2$.

      Similar devices can be made for other sentential connectives. For a two-input OR gate (Fig. 2) the output voltage is the maximum of the input voltages. Corresponding to the negation connective there is the NOT device (or inverter), whose output voltage is the opposite of the input voltage.

      A circuit can be constructed from various devices of this sort. And it is again natural to use wffs of our formal language to label the voltages

---

[1] This section, which discusses an application of the ideas of previous sections, may be omitted without loss of continuity.
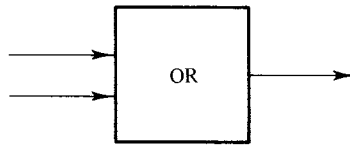
Figure 2. OR gate.

at different points (Fig. 3). Conversely, given the wff thus attached to the output, we can approximately reconstruct the circuit, which looks very much like the tree of the wff's formation.
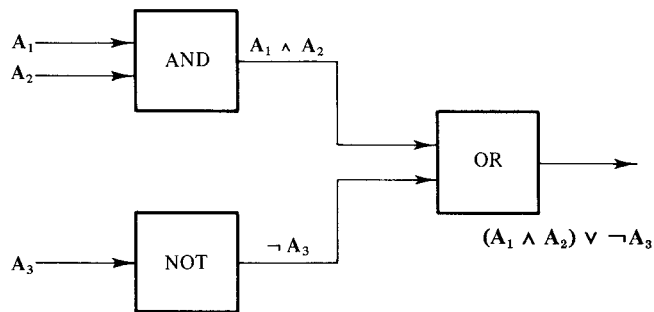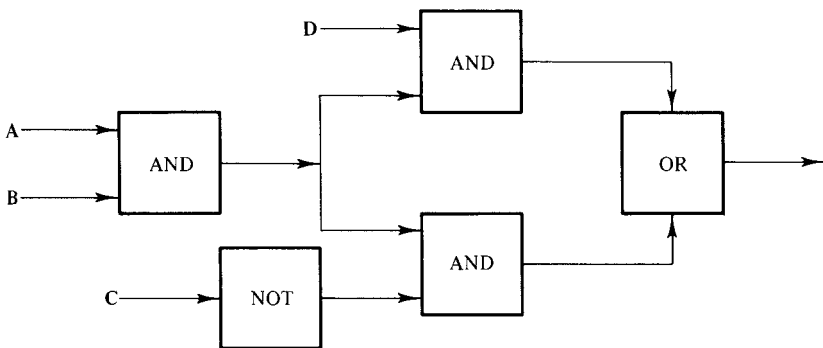

Figure 3. Circuit with wffs as labels.

For example, the circuit for

$$((A \land B) \land D) \lor ((A \land B) \land \neg C)$$

would probably be as shown in Fig. 4. Duplication of the circuit for **A** $\land$ **B** would not usually be desirable.

Figure 4. Circuit for **$((A \land B) \land D) \lor ((A \land B) \land \neg C)$**.

Tautologically equivalent wffs yield circuits having ultimately the

same performance, although possibly at different cost and (if the devices are not quite instantaneous in operation) different speed. Define the *delay* (also called the *depth*) of a circuit as the maximum number of boxes through which the signal can pass in going from an input to the output. The corresponding notion for formulas is conveniently defined by recursion.

1. The delay of a sentence symbol is 0.
2. The delay of $\neg\alpha$ is one greater than the delay of α.
3. The delay of $\alpha \wedge \beta$ is one greater than the maximum of the delay of α and the delay of β.

And similarly for any other connective.

For example, the circuit of $(\mathbf{A}_1 \wedge \mathbf{A}_2) \vee \neg\, \mathbf{A}_3$ uses three devices and has a delay of 2. The tautologically equivalent formula $\neg\,(\mathbf{A}_3 \wedge (\neg\, \mathbf{A}_1 \vee \neg\, \mathbf{A}_2))$ gives a circuit having five devices and a delay of 4. The problem facing many a computer engineer can be stated: Given a circuit (or its wff), find an equivalent circuit (or a tautologically equivalent wff) for which the cost is a minimum, subject to constraints such as a maximum allowable delay. For this problem there is some catalog of available devices; for example, she might have available

$$\text{NOT, two-input AND, three-input OR.}$$

(It is clearly desirable that the available devices correspond to a complete set of connectives.) The catalog of devices determines a formal language, having a connective symbol for each device.

EXAMPLE 1.   Inputs: **A**, **B**, **C**. Output: To agree with the majority of **A**, **B**, and **C**. Devices available: two-input OR, two-input AND. One solution is

$$((\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C})) \vee (\mathbf{B} \wedge \mathbf{C}),$$

which uses five devices and has a delay of 3. But a better solution is

$$(\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C})) \vee (\mathbf{B} \wedge \mathbf{C}),$$

which uses four devices and has the same delay. Furthermore, there is no solution using only three devices; cf. Exercise 1.

EXAMPLE 2.   Inputs: **A** and **B**. Output: T if the inputs agree, F otherwise; i.e., the circuit is to test for equality. Device available: two-input NOR. One solution is

$$((\mathbf{A} \downarrow \mathbf{A}) \downarrow \mathbf{B}) \downarrow ((\mathbf{B} \downarrow \mathbf{B}) \downarrow \mathbf{A}).$$

This uses five devices; is there a better solution? And there is

the deeper question: Is there an efficient procedure for finding a minimal solution? These are questions that we merely raise here. In recent years a great deal of work has gone into investigating questions of this type.

EXAMPLE 3.   (Relay switching) Inputs: **A**, ¬**A**, **B**, ¬**B**, . . . Devices: OR (any number of inputs), AND (any number of inputs). Cost: Devices are free, but each use of an input costs one unit. To test for equality of **A** and **B** we could use

$$(\mathbf{A} \wedge \mathbf{B}) \vee (\neg\,\mathbf{A} \wedge \neg\,\mathbf{B}).$$

The wiring diagram for the circuit is shown in Fig. 5. The circuit will pass current iff **A** and **B** are assigned the same value. (This formula, equivalent to $\mathbf{A} \leftrightarrow \mathbf{B}$, has the property that its truth value changes whenever the value of one argument changes. For this reason, the circuit is used, with double-throw switches, in wiring hallway lights.)
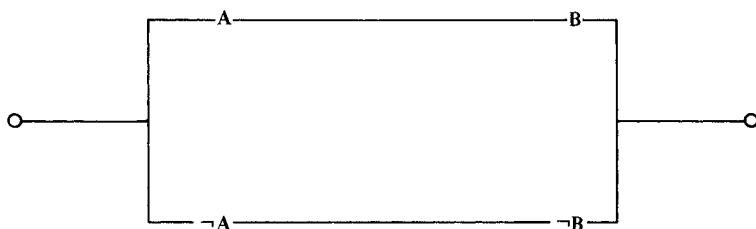

Figure 5. Wiring diagram for $(\mathbf{A} \wedge \mathbf{B}) \vee (\neg\mathbf{A} \wedge \neg\mathbf{B})$.

But there is one respect in which relay circuits do not fit the description given at the beginning of this section. Relays are bilateral devices; they will pass current in either direction. This feature makes "bridge" circuits possible (Fig. 6). The methods described here do not apply to such circuits.
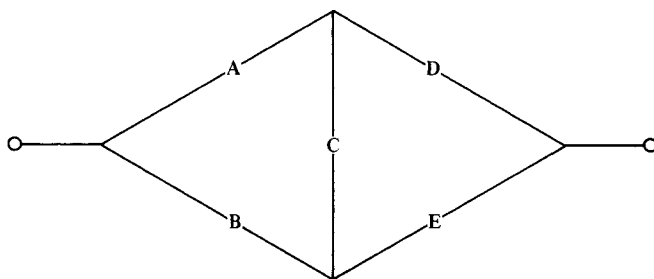

Figure 6. Bridge circuit.

EXAMPLE 4.   There are four inputs, and the circuit is to realize the Boolean function G, where G is to have the value T at $\langle F, F, F, T \rangle, \langle F, F, T, T \rangle, \langle F, F, T, F \rangle, \langle F, F, T, T \rangle, \langle F, T, F, F \rangle, \langle F, T, F, T \rangle, \langle F, T, T, F \rangle, \langle F, T, T, T \rangle$, and $\langle T, F, F, T \rangle$. G is to have the value F at $\langle T, F, F, F \rangle, \langle T, F, T, F \rangle, \langle T, T, F, F \rangle, \langle T, T, T, F \rangle$, and $\langle T, T, T, T \rangle$. At the remaining three points, $\langle F, F, F, F \rangle, \langle T, F, T, T \rangle$, and $\langle T, T, F, T \rangle$, we don't care about the value of G. (The application of the circuit is such that these three combinations never occur.)

We know that G can be realized by using, say, $\{\wedge, \vee, \neg\}$. But we want to do this in an efficient way. The first step is to represent the data in a more comprehensible form. We can do this by means of Fig. 7. Since $G(F, F, F, T) = T$, we have placed a T in the square with coordinates $\langle \neg \mathbf{A}, \neg \mathbf{B}, \neg \mathbf{C}, \mathbf{D} \rangle$. Similarly, there is an F in the square with coordinates $\langle \mathbf{A}, \mathbf{B}, \neg \mathbf{C}, \neg \mathbf{D} \rangle$ because $G(T, T, F, F) = F$. The three squares we do not care about are left blank.
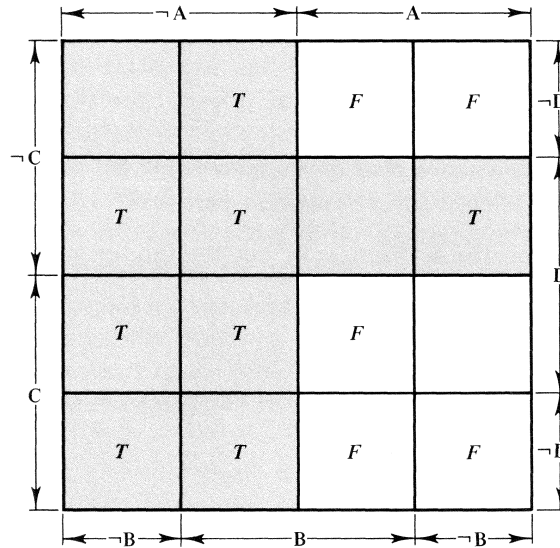

Figure 7. Diagram for Example 4.

Now we look for a simple geometrical pattern. The shaded area includes all T's and no F's. It corresponds to the formula

$$(\neg \mathbf{A}) \vee (\neg \mathbf{C} \wedge \mathbf{D}),$$

which is reasonably simple and meets all our requirements. Note that the input **B** is not needed at all.

Exercises

1. In Example 1 of this section, verify that there is no solution using only three devices.

2. Define a *literal* to be a wff which is either a sentence symbol or the negation of a sentence symbol. An *implicant* of φ is a conjunction α of literals (using distinct sentence symbols) such that $\alpha \models \varphi$. We showed in Section 1.5 (cf. Corollary 15C) that any satisfiable wff φ is tautologically equivalent to a disjunction $\alpha_1 \vee \cdots \vee \alpha_n$ where each α_i is an implicant of φ. An implicant α of φ is *prime* iff it ceases to be an implicant upon the deletion of any of its literals. Any disjunction of implicants equivalent to φ clearly must, if it is to be of minimum length, consist only of prime implicants.

    (a) Find all prime implicants of

$$(A \rightarrow B) \wedge (\neg A \rightarrow C).$$

    (b) Which disjunctions of prime implicants enjoy the property of being tautologically equivalent to the formula in part (a)?

3. Repeat (a) and (b) of Exercise 2, but for the formula

$$(A \vee \neg B) \wedge (\neg C \vee D) \rightarrow B \wedge ((A \wedge C) \vee (\neg C \wedge D)).$$

SECTION 1.7

Compactness and Effectiveness

Compactness

We now give a proof of the compactness theorem mentioned earlier (Section 1.2). Call a set Σ of wffs *satisfiable* iff there is a truth assignment that satisfies every member of Σ.

> **COMPACTNESS THEOREM**   A set of wffs is satisfiable iff every finite subset is satisfiable.

Let us temporarily say that Σ is *finitely satisfiable* iff every finite subset of Σ is satisfiable. Then the compactness theorem asserts that this notion coincides with satisfiability. Notice that if Σ is satisfiable, then automatically it is finitely satisfiable. Also if Σ is finite, then the converse is trivial. (Every set is a subset of itself.) The nontrivial part is to show that if an infinite set is finitely satisfiable, then it is satisfiable.

> PROOF OF THE COMPACTNESS THEOREM.   The proof consists of two distinct parts. In the first part we take our given finitely satisfiable set Σ and extend it to a maximal such set Δ. In the second part we utilize Δ to make a truth assignment that satisfies Σ.

For the first part let $\alpha_1, \alpha_2, \ldots$ be a fixed enumeration of the wffs. (This is possible since the set of sentence symbols, and hence the set of expressions, is countable; see Theorem 0B.) Define by recursion (on the natural numbers)

$$\Delta_0 = \Sigma,$$
$$\Delta_{n+1} = \begin{cases} \Delta_n; \alpha_{n+1} & \text{if this is finitely satisfiable,} \\ \Delta_n; \neg\alpha_{n+1} & \text{otherwise.} \end{cases}$$

(Recall that $\Delta_n; \alpha_{n+1} = \Delta_n \cup \{\alpha_{n+1}\}$.) Then each Δ_n is finitely satisfiable; see Exercise 1. Let $\Delta = \bigcup_n \Delta_n$, the limit of the Δ_n's.

It is clear that (1) $\Sigma \subseteq \Delta$ and that (2) for any wff α either $\alpha \in \Delta$ or $(\neg\alpha) \in \Delta$. Furthermore, (3) Δ is finitely satisfiable. For any finite subset is already a finite subset of some Δ_n and hence is satisfiable.

This concludes the first part of the proof; we now have a set Δ having properties (1)–(3). There is in general not a unique such set, but there is at least one. (An alternative proof of the existence of such a Δ — and one that we can use even if there are uncountably many sentence symbols — employs Zorn's lemma. The reader familiar with uses of Zorn's lemma should perceive its applicability here.)

For the second part of the proof we define a truth assignment v for the set of all sentence symbols:

$$v(A) = T \quad \text{iff} \quad A \in \Delta$$

for any sentence symbol A. Then for any wff φ, we claim that

$$v \text{ satisfies } \varphi \quad \text{iff} \quad \varphi \in \Delta.$$

This is proved by induction on φ; see Exercise 2. Since $\Sigma \subseteq \Delta$, v must then satisfy every member of Σ.   ⊣

COROLLARY 17A   If $\Sigma \models \tau$, then there is a finite $\Sigma_0 \subseteq \Sigma$ such that $\Sigma_0 \models \tau$.

PROOF.   We use the basic fact that $\Sigma \models \tau$ iff $\Sigma; \neg\tau$ is unsatisfiable.

$\Sigma_0 \not\models \tau$ for every finite $\Sigma_0 \subseteq \Sigma$
    $\Rightarrow \Sigma_0; \neg\tau$ is satisfiable for every finite $\Sigma_0 \subseteq \Sigma$
    $\Rightarrow \Sigma; \neg\tau$ is finitely satisfiable
    $\Rightarrow \Sigma; \neg\tau$ is satisfiable
    $\Rightarrow \Sigma \not\models \tau.$
                                                  ⊣

In fact, the above corollary is equivalent to the compactness theorem; see Exercise 3.

Effectiveness and Computability

Although the method of truth tables is rather cumbersome to use, the existence of the method yields interesting theoretical conclusions. Suppose we ask of a set Σ of wffs whether or not there is an *effective* procedure that, given a wff τ, will decide whether or not $\Sigma \models \tau$. By an *effective* procedure we mean one meeting the following conditions:

    1. There must be exact instructions (i.e., a *program*) explaining how to execute the procedure. The instructions must be finite in length. After all, it must be possible to *give* the instructions to the person or machine doing the calculations, and one cannot give someone all of an infinite object. But we impose no upper limit in advance on exactly how long the instructions can be. If there are more lines in the instructions than there are electrons in the universe, we merely shrug and say, "That's a pretty long program."

    2. These instructions must not demand any brilliance or originality on the part of the person or machine following them. The idea is that a diligent clerk (who knows no mathematics but is very good at following directions) or your computing machine (which does not think at all) should be able to execute the instructions. That is, it must be possible for the instructions to be *mechanically implemented*. The procedure must avoid random devices (such as the flipping of a coin), or any such device that can, in practice, only be approximated.

    3. In the case of a decision procedure, as asked for above, the procedure must be such that, given a wff τ, after a finite number of steps the procedure produces a "yes" or "no" answer. (That is, the procedure must be an *algorithm* for determining the answer.)

    On the other hand, we place no bound in advance on the amount of time that might be required before the answer appears. Nor do we place any advance bound on the amount of scratch paper (or other storage medium) that might be required. These will depend on, among other things, the input τ. But for any one τ, the procedure is to require only a finite number of steps to produce the answer, and so only a finite amount of scratch paper will be consumed. It is no fair doing an infinite number of steps and *then* giving the answer.

    People with a digital computing machine may regard a procedure as being effective only when it can be carried out on their machine in a "reasonable" length of time. Of course, the matter of what is reasonable may change with the circumstances. Next year they plan to get a faster machine with more memory. At that time, their idea of what can be done in a reasonable length of time will be considerably extended. We want here a concept of effective procedure that is the limiting case where all the practical restrictions on running time and memory space are removed.

Of course the foregoing description can hardly be considered a precise definition of the word "effective." And, in fact, that word will be used only in an informal intuitive way throughout this book. (In Chapter 3 we will meet a precise counterpart, "recursive.") But as long as we restrict ourselves to positive assertions that there *does* exist an effective procedure of a certain sort, the informal approach suffices. We simply display the procedure, show that it works, and people will agree that it is effective. (But this relies on the *empirical* fact that procedures which appear effective to one mathematician also appear so to others.) If we wanted a negative result, that there did *not* exist an effective procedure of a certain sort, then this informal viewpoint would be inadequate. (In Chapter 3 we do want to obtain just such negative results.) Because the notion of effectiveness is informal, definitions and theorems involving it will be marked with a star. For example:

⋆**THEOREM 17B**   There is an effective procedure that, given any expression ε, will decide whether or not it is a wff.

PROOF.   See the algorithm in Section 1.3 and the footnotes thereto.

⊣

There is one technical point here, stemming from the fact that our language has infinitely many different sentence symbols. When we speak of being "given" an expression ε, we imagine that someone might write down the symbols in ε, one after the other. It is implausible that anyone has the potential of writing down any one of an infinitude of symbols. To avoid this, we adopt the following "input/output format": Instead of writing down $\mathbf{A}_5$, for example, we use $\mathbf{A}''''$, a string of five symbols. Now the total number of symbols in our alphabet is only nine:

$$(,), \; \neg, \; \wedge, \; \vee, \; \rightarrow, \; \leftrightarrow, \; \mathbf{A}, \quad \text{and} \quad '.$$

(If we identify these nine symbols with the digits 1–9, we get expressions that look particularly familiar in computational settings! And we still have the 0 digit for separating expressions.)

Theorem 17B states that the set of wffs is *decidable* in the sense of the following definition:

⋆DEFINITION.   A set Σ of expressions is *decidable* iff there exists an effective procedure that, given an expression α, will decide whether or not $\alpha \in \Sigma$.

For example, any finite set is decidable. (The instructions can simply list the finitely many members of the set. Then the algorithm can check the input against the list.) Some infinite sets are decidable but not all. On the one hand, there are uncountably many ($2^{\aleph_0}$, to be exact) sets of expressions. On the other hand, there can be only countably many effective procedures. This is because the procedure is completely deter-

mined by its (finite) instructions. There are only $\aleph_0$ finite sequences of letters, and the instructions, when written out, form a finite sequence of letters.

   *THEOREM 17C   There is an effective procedure that, given a finite set $\Sigma; \tau$ of wffs, will decide whether or not $\Sigma \models \tau$.

   PROOF.   The truth-table procedure (Section 1.2) meets the requirement.   ⊣

   In this theorem we specified that $\Sigma; \tau$ was finite, since one cannot be "given" in any direct and effective way all of an infinite object.

   *COROLLARY 17D   For a finite set Σ, the set of tautological consequences of Σ is decidable. In particular, the set of tautologies is decidable.

   If Σ is an infinite set — even a decidable one — then in general its set of tautological consequences may not be decidable. (See Chapter 3.) But we can obtain a weaker result, which is in a sense half of decidability.

   Say that a set A of expressions is *effectively enumerable* iff there exists an effective procedure that lists, in some order, the members of A. If A is infinite, then the procedure can never finish. But for any specified member of A, it must eventually (i.e., in a finite length of time) appear on the list.

   To give more of a feeling for this concept, we now state an equivalent way of formulating it.

   *THEOREM 17E   A set A of expressions is effectively enumerable iff there exists an effective procedure that, given any expression ε, produces the answer "yes" iff $\varepsilon \in A$.

   If $\varepsilon \notin A$, the procedure might produce the answer "no"; more likely it will go on forever without producing any answer, but it must not lie to us and produce the answer "yes." Such a procedure is called a *semidecision* procedure — it is half of a decision procedure:

   *DEFINITION.   A set A of expressions is *semidecidable* iff there exists an effective procedure that, given any expression ε, produces the answer "yes" iff $\varepsilon \in A$.

   Thus Theorem 17E states that a set is effectively enumerable iff it is semidecidable.

   PROOF.   If A is effectively enumerable, then given any ε we can examine the listing of A as our procedure churns it out. If and when ε appears, we say "yes." (Thus if $\varepsilon \notin A$, no answer is ever given. It is this that keeps A from being decidable. When ε has failed to occur among the first 10^{10} enumerated members of A, there is in general no way of knowing whether $\varepsilon \notin A$ — in which

case one should give up looking — or whether ε will occur in the very next step.)

Conversely, suppose that we have the procedure described in the theorem. We want to create a listing of A. The idea is to enumerate all expressions, and to apply our given procedure to each. But we must budget our time sensibly. It is easy enough to enumerate effectively all expressions:

$$\varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots.$$

Then proceed according to the following scheme:

1. Spend one minute testing ε_1, for membership in A (using the given procedure).
2. Spend two minutes testing ε_1, then two minutes testing ε_2.
3. Spend three minutes testing ε_1, three minutes testing ε_2, and three minutes testing ε_3.

And so forth. Of course whenever our procedure produces a "yes" answer, we put the accepted expression on the output list. Thus any member of A will eventually appear on the list. (It will appear infinitely many times, unless we modify the above instructions to check for duplication.)   ⊣

Clearly any decidable set is also semidecidable. (We get a semidecision procedure even if the "no" bulb has burned out.) Hence any any decidable set is effectively enumerable.

⋆**THEOREM 17F**   A set of expressions is decidable iff both it and its complement (relative to the set of all expressions) are effectively enumerable.

PROOF.   Exercise 8. This result is sometimes called "Kleene's theorem."   ⊣

Observe that if sets A and B are effectively enumerable, then so are $A \cup B$ and $A \cap B$ (Exercise 11). The class of decidable sets is also closed under union and intersection, and it is in addition closed under complementation.

Now for a more substantive result:

⋆**THEOREM 17G**   If Σ is a decidable set of wffs, then the set of tautological consequences of Σ is effectively enumerable.

PROOF.   Actually it is enough for Σ to be effectively enumerated; consider an enumeration

$$\sigma_1, \sigma_2, \sigma_3, \ldots.$$

Given any wff τ, we can test (by truth tables) successively whether or not

$$\varnothing \models \tau,$$
$$\{\sigma_1\} \models \tau,$$
$$\{\sigma_1, \sigma_2\} \models \tau,$$
$$\{\sigma_1, \sigma_2, \sigma_3\} \models \tau,$$

and so forth. If any of these conditions is met, then we answer "yes." Otherwise, we keep trying.

This does produce an affirmative answer whenever $\Sigma \models \tau$, by the corollary to the compactness theorem. $\dashv$

Later on, we will want to use effective procedures to compute functions. We will say that a function f is *effectively computable* (or simply *computable*) iff there exists an effective procedure that, given an input x, will eventually produce the correct output $f(x)$.

Exercises

1. Assume that every finite subset of Σ is satisfiable. Show that the same is true of at least one of the sets $\Sigma; \alpha$ and $\Sigma; \neg\alpha$. (This is part of the proof of the compactness theorem.) *Suggestion*: If not, then $\Sigma_1; \alpha$ and $\Sigma_2; \neg\alpha$ are unsatisfiable for some finite $\Sigma_1 \subseteq \Sigma$ and $\Sigma_2 \subseteq \Sigma$. Look at $\Sigma_1 \cup \Sigma_2$.

2. Let Δ be a set of wffs such that (i) every finite subset of Δ is satisfiable, and (ii) for every wff α, either $\alpha \in \Delta$ or $(\neg\alpha) \in \Delta$. Define the truth assignment v:

$$v(A) = \begin{cases} T & \text{iff} \quad A \in \Delta, \\ F & \text{iff} \quad A \notin \Delta \end{cases}$$

for each sentence symbol A. Show that for every wff φ, $\bar{v}(\varphi) = T$ iff $\varphi \in \Delta$. (This is part of the proof of the compactness theorem.) *Suggestion*: Use induction on φ.

3. Show that from the corollary to the compactness theorem we can prove the compactness theorem itself (far more easily than we can starting from scratch).

4. In 1977 it was proved that every planar map can be colored with four colors. Of course, the definition of "map" requires that there be only finitely many countries. But extending the concept, suppose we have an infinite (but countable) planar map with countries $C_1, C_2, C_3, \ldots$. Prove that this infinite planar map can still be colored with four colors. (Suggestion: Partition the sentence symbols into four parts. One sentence symbol, for example, can be used to translate, "Country C_7 is colored red." Form a set Σ_1 of wffs that say, for example, C_7 is exactly one of the colors. Form another set

Σ_2 of wffs that say, for each pair of adjacent countries, that they are not the same color. Apply compactness to $\Sigma_1 \cup \Sigma_2$.)

5. Where Σ is a set of wffs, define a *deduction* from Σ to be a finite sequence $\langle \alpha_0, \ldots, \alpha_n \rangle$ of wffs such that for each $k \leq n$, either (a) α_k is a tautology, (b) $\alpha_k \in \Sigma$, or (c) for some i and j less than k, α_i is $(\alpha_j \to \alpha_k)$. (In case (c), one says that α_k is obtained by *modus ponens* from α_i and α_j.) Give a deduction from the set

$$\{ \neg \mathbf{S} \vee \mathbf{R}, \mathbf{R} \to \mathbf{P}, \mathbf{S} \}$$

the last component of which is $\mathbf{P}$.

6. Let $\langle \alpha_0, \ldots, \alpha_n \rangle$ be a deduction from a set Σ of wffs, as in the preceding problem. Show that $\Sigma \models \alpha_k$ for each $k \leq n$. *Suggestion*: Use strong induction on k, so that the inductive hypothesis is that $\Sigma \models \alpha_i$ for all $i < k$.

7. Show that whenever $\Sigma \models \tau$, then there exists a deduction from Σ, the last component of which is τ. *Remark*: This result is called "completeness"; the concepts in Exercises 5–7 will reappear in Section 2.4.

8. Prove Theorem 17F. *Remark*: Two semidecision procedures make a whole.

*9. The concepts of decidability and effective enumerability can be applied not only to sets of expressions but also to sets of integers or to sets of pairs of expressions or integers. Show that a set A of expressions is effectively enumerable iff there is a decidable set B of pairs $\langle \alpha, n \rangle$ (consisting of an expression α and an integer n) such that $A = \operatorname{dom} B$.

10. Let Σ be an effectively enumerable set of wffs. Assume that for each wff τ, either $\Sigma \models \tau$ or $\Sigma \models \neg \tau$. Show that the set of tautological consequences of Σ is decidable.
    (a) Do this where "or" is interpreted in the exclusive sense: either $\Sigma \models \tau$ or $\Sigma \models \neg \tau$ but not both.
    (b) Do this where "or" is interpreted in the inclusive sense: either $\Sigma \models \tau$ or $\Sigma \models \neg \tau$ or both.

11. (a) Explain why the union of two effectively enumerable sets is again effectively enumerable.
    (b) Explain why the intersection of two effectively enumerable sets is again effectively enumerable.

12. For each of the following conditions, give an example of an *unsatisfiable* set Γ of formulas that meets the condition.
    (a) Each member of Γ is — by itself — satisfiable.
    (b) For any *two* members γ_1 and γ_2 of Γ, the set $\{\gamma_1, \gamma_2\}$ is satisfiable.
    (c) For any *three* members γ_1, γ_2, and γ_3 of Γ, the set $\{\gamma_1, \gamma_2, \gamma_3\}$ is satisfiable.

Chapter TWO

First-Order Logic

SECTION 2.0
Preliminary Remarks

The preceding chapter presented the first of our mathematical models of deductive thought. It was a simple model, indeed *too* simple. It is easy to think of examples of intuitively correct deductions that cannot be adequately mirrored in the model of sentential logic.

Suppose we begin with a collection of hypotheses (in English) and a possible conclusion. By translating everything to the language of sentential logic we obtain a set Σ of hypotheses and a possible conclusion τ. Now if $\Sigma \models \tau$, then we feel that the original English-language deduction was valid. But if $\Sigma \not\models \tau$, then we are unsure. It may well be that the model of sentential logic was simply too crude to mirror the subtlety of the original deduction.

This chapter presents a system of logic of much greater ability. In fact, when the "working mathematician" finds a proof, almost invariably what is meant is a proof that can be mirrored in the system of this chapter.

First, we want to give an informal description of the features our first-order languages might have (or at least might be able to simulate). We begin with a special case, the first-order language for number theory. For this language there is a certain intended way of translating to and from English (Table VII).

TABLE VII

Formal expression	Intended translation
$\mathbf{0}$	"Zero." Here $\mathbf{0}$ is a constant symbol, intended to name the number 0.
$\mathbf{S}t$	"The successor of t." Here $\mathbf{S}$ is a one-place function symbol. t is to be an expression that names some number a. Then $\mathbf{S}t$ names $S(a)$, the successor of a. For example, $\mathbf{S0}$ is intended to name the number 1.
$< v_1 v_2$	"v_1 is less than v_2." Here $<$ is a two-place predicate symbol. At the end of Section 2.1 we will adopt conventions letting us abbreviate the expression in the more usual style: $v_1 < v_2$.
$\forall$	"For every natural number." The symbol $\forall$ is the universal quantifier symbol. More generally, with each translation of the language into English there will be associated a certain set A (the so-called universe); $\forall$ will then become "for every member of the universe A."
$\forall\, v_1 < \mathbf{0} v_1$	"For every natural number v_1, zero is less than v_1." Or more euphoniously, "Every natural number is larger than 0." This formal sentence is false in the intended translation, since zero is not larger than itself.

One abbreviation is mentioned in Table VII. There will be more (Table VIII).

Actually we will not be quite as generous as the tables might suggest. There are two economy measures that we can take to obtain simplification without any essential loss of linguistic expressiveness:

First, we choose as our sentential connective symbols just $\neg$ and $\rightarrow$. We know from Section 1.5 that these form a complete set, so there is no compelling reason to use more.

Secondly, we forego the luxury of an existential quantifier, $\exists\, x$. In its place we use $\neg\, \forall\, x\, \neg$. This is justified, since an English sentence like

There is something rotten in the state of Denmark,

is equivalent to

It is not the case that for every x, x is not rotten in the state of Denmark.

Thus the formula $\exists\, v_1 \,\forall\, v_2\, v_1 = v_2$ becomes, in unabbreviated form,

$$(\neg\, \forall\, v_1 (\neg\, \forall\, v_2 = v_1 v_2)).$$

TABLE VIII

Abbreviated expression	Intended translation
$x = y$	"x equals y." In unabbreviated form this will become $= xy$.
$\exists\, v$	"There exists a natural number v such that." Or more generally, "there exists a member of the universe such that."
$\exists\, v_1\, \forall\, v_2\, v_1 = v_2$	"There is exactly one natural number." Again this formal sentence is false in the intended translation.
$\forall\, v_1(0 < v_1 \lor 0 = v_1)$	"Every natural number is greater than or equal to zero."

For an example in an ad hoc language, we might translate "Socrates is a man" as Hs, where H is a one-place predicate symbol intended to translate "is a man" and s is a constant symbol intended to name Socrates. Similarly, to translate "Socrates is mortal" we take Ms. Then "All men are mortal" is translated as $\forall\, v_1(H v_1 \rightarrow M v_1)$.

The reader will possibly recognize the symbols $\forall$ and $\exists$ from previous mathematical contexts. Indeed, some mathematicians, when writing on the blackboard during their lectures, already use a nearly formalized language with only vestigial traces of English. That our first-order languages resemble theirs is no accident. We want to be able to take one step back and study not, say, sets or groups, but the sentences of set theory or group theory. (The term *metamathematics* is sometimes used; the word itself suggests the procedure of stepping back and examining what the mathematician is doing.) The objects you, the logician, now study are the sentences that you, the set theoretician, previously used in the study of sets. This requires formalizing the language of set theory. And we want our formal languages to incorporate the features used in, for example, set theory.

SECTION 2.1
First-Order Languages

We assume henceforth that we have been given infinitely many distinct objects (which we call symbols), arranged as follows:

   A. Logical symbols
      0. Parentheses: (,).
      1. Sentential connective symbols: $\rightarrow$, $\neg$.

2. Variables (one for each positive integer n):

$$v_1, \, v_2, \, \ldots.$$

3. Equality symbol (optional): $=$.

B. Parameters

   0. Quantifier symbol: $\forall$.
   1. Predicate symbols: For each positive integer n, some set (possibly empty) of symbols, called n-place predicate symbols.
   2. Constant symbols: Some set (possibly empty) of symbols.
   3. Function symbols: For each positive integer n, some set (possibly empty) of symbols, called n-place function symbols.

In A.3 we allow for the possibility of the equality symbol's being present, but we do not assume its presence. Some languages will have it and others will not. The equality symbol is a two-place predicate symbol but is distinguished from the other two-place predicate symbols by being a logical symbol rather than a parameter. (This status will affect its behavior under translations into English.) We do need to assume that some n-place predicate symbol is present for some n.

In B.2, the constant symbols are also called 0-place function symbols. This will often allow a uniform treatment of the symbols in B.2 and B.3.

As before, we assume that the symbols are distinct and that no symbol is a finite sequence of other symbols.

In order to specify which language we have before us (as distinct from other first-order languages), we must (i) say whether or not the equality symbol is present, and (ii) say what the parameters are.

We now list some examples of what this language might be:

1. *Pure predicate language*

Equality: No.
n-place predicate symbols: $A_1^n, A_2^n, \ldots.$
Constant symbols: $a_1, a_2, \ldots.$
n-place function symbols ($n > 0$): None.

2. *Language of set theory*

Equality: Yes (usually).
Predicate parameters: One two-place predicate symbol $\in$.
Function symbols: None (or occasionally a constant symbol $\varnothing$).

3. *Language of elementary number theory* (as in Chapter 3)

Equality: Yes.
Predicate parameters: One two-place predicate symbol $<$.
Constant symbols: The symbol **0**.

One-place function symbols: **S** (for successor).

Two-place function symbols: $+$ (for addition), $\cdot$ (for multiplication), and **E** (for exponentiation).

In examples 2 and 3 there are certain intended translations of the parameters. We will presently give a number of examples of sentences that can be translated into these languages and a few examples of sentences that cannot be so translated.

It is important to notice that our notion of language includes the language for set theory. For it is generally agreed that, by and large, mathematics can be embedded into set theory. By this is meant that

(a) Statements in mathematics (such as the fundamental theorem of calculus) can be expressed in the language of set theory; and

(b) The theorems of mathematics follow logically from the axioms of set theory.

Our model of first-order logic is fully adequate to mirror this procedure.

EXAMPLES in the language of set theory. Here it is intended that $\forall$ should mean "for all sets" and $\in$ should mean "is a member of."

1. "There is no set of which every set is a member." We will translate this into the language of set theory using several steps. The intermediate sentences are neither in English nor in the formal language but are in a mixed language.

$\neg$[There is a set of which every set is a member]

$\neg \exists\, v_1$[Every set is a member of v_1]

$\neg \exists\, v_1 \forall\, v_2\ v_2 \in v_1$

Although it is tempting to stop here, we must now replace $v_2 \in v_1$ by $\in v_2 v_1$, since predicate symbols will always go at the left in such contexts. Furthermore, $\exists\, v_1$ must be replaced by $\neg \forall\, v_1\, \neg$, as mentioned earlier. And we must use the correct number of parentheses. The finished product is

$$(\neg\,(\neg \forall\, v_1 (\neg \forall\, v_2\ \in v_2 v_1))).$$

2. Pair-set axiom: "For any two sets, there is a set whose members are exactly the two given sets." Again we carry out the translation in stages.

$\forall\, v_1 \forall\, v_2$ [There is a set whose members are exactly v_1 and v_2]

$\forall\, v_1 \forall\, v_2 \exists\, v_3$ [The members of v_3 are exactly v_1 and v_2]

$\forall\, v_1 \forall\, v_2 \exists\, v_3 \forall\, v_4 (v_4 \in v_3 \leftrightarrow v_4 = v_1 \vee v_4 = v_2)$

Now we replace $\exists\, v_3$ by $\neg \forall\, v_3\, \neg$, $v_4 \in v_3$ by $\in v_4 v_3$, and $v_4 = v_i$ by $= v_4 v_i$. In addition, we must eliminate $\leftrightarrow$ and $\wedge$ in favor of our chosen connectives $\to$ and $\neg$. Thus

$$\alpha \vee \beta \ \text{ becomes } \ \neg \alpha \to \beta;$$
$$\alpha \leftrightarrow \beta \ \text{ becomes } \ \neg\,((\alpha \to \beta) \to \neg\,(\beta \to \alpha)).$$

The finished product is

$$\forall\, v_1 \,\forall\, v_2(\neg\, \forall\, v_3(\neg\, \forall\, v_4(\neg\, ((\in v_4 v_3 \rightarrow ((\,\neg = v_4 v_1) \rightarrow$$
$$= v_4 v_2)) \rightarrow (\neg\, (((\neg = v_4 v_1) \rightarrow\, = v_4 v_2) \rightarrow\, \in v_4 v_3)))))).$$

The finished product is not nearly as pleasant to read as the version that preceded it. As we have no interest in deliberately making life unpleasant for ourselves, we will eventually adopt conventions allowing us to avoid seeing the finished product at all. But for the moment it should be regarded as an interesting, even if unattractive, novelty.

EXAMPLES in the language of elementary number theory. Here it is intended that $\forall$ should mean "for all natural numbers" and that $<, \mathbf{0}, \mathbf{S}, +, \cdot$, and $\mathbf{E}$ should have the obvious meanings.

1. As a name for the natural number 2 we have the term $\mathbf{SS0}$, since 2 is the successor of the successor of zero. Similarly, for 4 we have the term $\mathbf{SSSS0}$. For the phrase "$2 + 2$" it is tempting to use $\mathbf{SS0} + \mathbf{SS0}$. But we will adopt the policy of always putting the function symbol at the left (i.e., we will use Polish notation for function symbols). Thus corresponding to the English phrase "$2 + 2$" we have the term $+\,\mathbf{SS0}\,\mathbf{SS0}$. The English sentence "Two plus two is four" is translated as

$$=\, +\; \mathbf{SS0}\,\mathbf{SS0}\,\mathbf{SSSS0}.$$

(The spaces are inserted to help the reader, but they do not constitute an official feature of the language.)

2. "Any nonzero natural number is the successor of some number." We will perform the translation in three steps:

$\forall\, v_1\, [\,\text{If } v_1 \text{ is nonzero, then } v_1 \text{ is the successor of some number.}]$

$\forall\, v_1\, (v_1 \neq \mathbf{0} \rightarrow \exists\, v_2\; v_1 = \mathbf{S}v_2).$

$\forall\, v_1\, ((\neg = v_1 \mathbf{0}) \rightarrow (\neg\, \forall\, v_2(\neg = v_1 \mathbf{S}v_2))).$

3. "Any nonempty set of natural numbers has a least element." This cannot be translated into our language, because we cannot express "any ... set." This requires either something like the (first-order) language for set theory or a second-order language for number theory. We could, however, translate, "The set of primes has a least element." (The first step is to convert this sentence into, "There is a smallest prime." We leave the other steps to the reader; hints can be found in the next section.)

EXAMPLES in ad hoc languages

1. "All apples are bad."

$$\forall\, v_1(A v_1 \rightarrow B v_1).$$

2. "Some apples are bad."
Intermediate step: $\exists\, v_1(A\,v_1 \wedge B\,v_1)$.
Finished product: $(\neg\,\forall\, v_1(\neg\,(\neg\,(A\,v_1 \rightarrow (\neg\, B\,v_1)))))$.

These two examples illustrate patterns that arise continually. An English sentence which asserts that everything in a certain category has some property is translated

$$\forall\, v(\,\underline{} \rightarrow \underline{}\,).$$

A sentence which asserts that there is some object or objects in the category and having the property is translated

$$\exists\, v(\,\underline{} \wedge \underline{}\,).$$

The reader should be cautioned against confusing the two patterns. For example,

$$\forall\, v_1(A\,v_1 \wedge B\,v_1)$$

translates "Everything is an apple and is bad," which is a much stronger assertion than the sentence in the first example. Similarly, $\exists\, v_1(A\,v_1 \rightarrow B\,v_1)$ translates "There is something which is bad, if it is an apple." This is a much weaker assertion than the sentence in the second example. It is true (vacuously) even if all apples are good, provided only that the world has something which is not an apple.

3. "Bobby's father can beat up the father of any other kid on the block." Establish a language in which $\forall$ is intended to mean "for all people," $K x$ is to mean "x is a kid on the block," b is to mean "Bobby," $B x y$ is to mean "x can beat up y," and $f x$ is to mean "the father of x." Then a translation is

$$\forall\, v_1(K\, v_1 \rightarrow ((\neg\,{=}v_1 b) \rightarrow B f b f\, v_1)).$$

4. In calculus, we learn the meaning of "the function f converges to L as x approaches a":

$$\forall\, \varepsilon(\varepsilon > 0 \rightarrow \exists\, \delta(\delta > 0 \wedge \forall\, x(|x - a| < \delta \rightarrow |f x - L| < \varepsilon))).$$

This *is*, apart from notational matters, a formula of the sort we are interested in, using a predicate symbol for ordering, function symbols for f, subtraction, and absolute values, and constant symbols for 0, a, and L.

Formulas

An *expression* is any finite sequence of symbols. Of course most expressions are nonsensical, but there are certain interesting expressions: the terms and the wffs.

The terms are the nouns and pronouns of our language; they are the expressions that can be interpreted as naming an object. The atomic formulas will be those wffs having neither connective nor quantifier symbols.

terms	atomic formulas	other wffs	all other expressions

$$\underbrace{\hspace{4cm}}_{\text{wffs}}$$

The terms are defined to be those expressions that can be built up from the constant symbols and the variables by prefixing the function symbols. To restate this in the terminology of Chapter 1, we define for each n-place function symbol f, an n-place term-building operation $\mathcal{F}_f$ on expressions:

$$\mathcal{F}_f(\varepsilon_1, \ldots, \varepsilon_n) = f\varepsilon_1 \cdots \varepsilon_n.$$

DEFINITION.   The set of *terms* is the set of expressions that can be built up from the constant symbols and variables by applying (zero or more times) the $\mathcal{F}_f$ operations.

If there are no function symbols (apart from the constant symbols), then the terms are just the constant symbols and the variables. In this case we do not need an inductive definition.

Notice that we use Polish notation for terms by placing the function symbol at the left. The terms do not contain parentheses or commas. We will later prove a unique readability result, showing that given a term, we can unambiguously decompose it.

The terms are the expressions that are translated as names of objects (noun phrases), in contrast to the wffs which are translated as assertions about objects.

Some examples of terms in the language of number theory are

$$+v_2\mathbf{S0},$$
$$\mathbf{SSSS0},$$
$$+\mathbf{E}v_1\mathbf{SS0}\ \mathbf{E}v_2\mathbf{SS0}.$$

The atomic formulas will play a role roughly analogous to that played by the sentence symbols in sentential logic. An *atomic formula* is an expression of the form

$$Pt_1 \cdots t_n,$$

where P is an n-place predicate symbol and $t_1, \ldots, t_n$ are terms.

For example, $=v_1v_2$ is an atomic formula, since $=$ is a two-place predicate symbol and each variable is a term. In the language of set theory we have the atomic formula $\in v_5 v_3$.

Notice that the atomic formulas are not defined inductively. Instead we have simply said explicitly just which expressions are atomic formulas.

The *well-formed formulas* are those expressions that can be built up from the atomic formulas by use (zero or more times) of the connective symbols and the quantifier symbol. We can restate this in the terminology of Chapter 1 by first defining some formula-building operations on expressions:

$$\mathcal{E}_\neg(\gamma) = (\neg\,\gamma),$$
$$\mathcal{E}_\rightarrow(\gamma, \delta) = (\gamma \rightarrow \delta),$$
$$\mathcal{Q}_i(\gamma) = \forall\, v_i\gamma.$$

DEFINITION.   The set of *well-formed formulas* (*wffs*, or just *formulas*) is the set of expressions that can be built up from the atomic formulas by applying (zero or more times) the operations $\mathcal{E}_\neg$, $\mathcal{E}_\rightarrow$, and $\mathcal{Q}_i$ ($i = 1, 2, \ldots$).

For example, on the one hand, $\neg\, v_3$ is not a wff. (Why?) On the other hand,

$$\forall\, v_1((\neg\,\forall\, v_3(\neg\,\in v_3\, v_1)) \rightarrow (\neg\,\forall\, v_2(\in v_2\, v_1 \rightarrow$$
$$(\neg\,\forall\, v_4(\in v_4\, v_2 \rightarrow (\neg\,\in v_4\, v_1))))))$$

is a wff, as is demonstrated by the following tree:
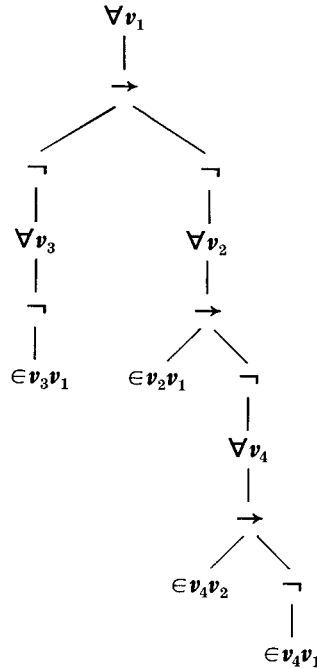
But it requires some study to discover that this wff is the axiom of regularity for set theory.

Free Variables

Two examples of wffs are $\forall v_2 \in v_2 v_1$ and $(\neg \forall v_1 (\neg \forall v_2 \in v_2 v_1))$. But there is an important difference between the two examples. The second might be translated back into English as

There is a set such that every set is a member of it.

The first example, however, can be translated only as an incomplete sentence, such as

Every set is a member of _____$_1$.

We are unable to complete the sentence without knowing what to do with v_1. In cases of this sort, we will say that v_1 *occurs free* in the wff $\forall v_2 \in v_2 v_1$. In contrast, no variable occurs free in $(\neg \forall v_1 (\neg \forall v_2 \in v_2 v_1))$. But of course we need a precise definition which does not refer to possible translations to English but refers only to the symbols themselves.

Consider any variable x. We define, for each wff α, what it means for x to *occur free* in α. This we do by recursion:

1. For atomic α, x occurs free in α iff x occurs in (i.e., is a symbol of) α.
2. x occurs free in $(\neg \alpha)$ iff x occurs free in α.
3. x occurs free in $(\alpha \rightarrow \beta)$ iff x occurs free in α or in β.
4. x occurs free in $\forall v_i \, \alpha$ iff x occurs free in α and $x \neq v_i$.

This definition makes tacit use of the recursion theorem. We can restate the situation in terms of functions. We begin with the function h defined on atomic formulas:

$h(\alpha) = $ the set of all variables, if any, in the atomic formula α.

And we want to extend h to a function $\overline{h}$ defined on all wffs in such a way that

$$\overline{h}(\mathcal{E}_\neg(\alpha)) = \overline{h}(\alpha),$$
$$\overline{h}(\mathcal{E}_\rightarrow(\alpha, \beta)) = \overline{h}(\alpha) \cup \overline{h}(\beta),$$
$$\overline{h}(\mathcal{Q}_i(\alpha)) = \overline{h}(\alpha) \text{ after removing } v_i, \text{ if present.}$$

Then we say that x occurs free in α (or that x is a *free variable* of α) iff $x \in \overline{h}(\alpha)$. The existence of a unique such $\overline{h}$ (and hence the meaningfulness of our definition) follows from the recursion theorem of Section 1.4 and from the fact (proved in Section 2.3) that each wff has a unique decomposition.

If no variable occurs free in the wff α (i.e., if $\bar{h}(\alpha) = \varnothing$), then α is a *sentence*. (The sentences are intuitively the wffs translatable into English without blanks, once we are told how to interpret the parameters.)

For example, $\forall\, v_2(A\,v_2 \rightarrow B\,v_2)$ and $\forall\, v_3(P\,v_3 \rightarrow \forall\, v_3\,Q\,v_3)$ are sentences, but v_1 occurs free in $(\,\forall\, v_1\,A\,v_1 \rightarrow B\,v_1)$. The sentences are usually the most interesting wffs. The others lead a second-class existence; they are used primarily as building blocks for sentences.

In translating a sentence from English, the choice of particular variables is unimportant. We earlier translated "All apples are bad" as $\forall\, v_1(A\,v_1 \rightarrow B\,v_1)$. We could equally well have used

$$\forall\, v_{27}(A\,v_{27} \rightarrow B\,v_{27}).$$

The variable is, in effect, used as a pronoun, just as in English we might say, "For any object whatsoever, if *it* is an apple, then *it* is bad." (We have incorporated into our language an adequate supply of pronouns: it_1, it_2, ...) Since the choice of particular variables is unimportant, we will often not even specify the choice. Instead we will write, for example, $\forall\, x(Ax \rightarrow Bx)$, where it is understood that x is some variable. (The unimportance of the choice of variable will eventually become a theorem.)

Similar usages of variables occur elsewhere in mathematics. In

$$\sum_{i=1}^{7} a_{ij}$$

i is a "dummy" variable but j occurs free.

On Notation

We can specify a wff (or indeed any expression) by writing a line that displays explicitly every symbol. For example,

$$\forall\, v_1((\neg = v_1 0) \rightarrow (\neg\forall\, v_2(\neg = v_1 S v_2))).$$

But this way of writing things, while splendidly complete, may not be readily comprehensible. The incomprehensibility can be blamed (in part) on the simplifications we wanted in the language (such as the lack of an existential quantifier symbol). We naturally want to have our cake and eat it too, so we now will agree on methods of specifying wffs in more indirect but more readable ways. These conventions will let us write a line such as

$$\forall\, v_1(v_1 \neq \mathbf{0} \rightarrow \exists\, v_2\, v_1 = S v_2)$$

to name the same wff as is named by the other line above.

Note well that we are *not* changing our definition of what a wff is. We are simply conspiring to fix certain ways of naming wffs. In the (rare) cases where the exact sequence of symbols becomes important,

we may have to drop these new conventions and revert to primitive notation.

We adopt then the following abbreviations and conventions. Here α and β are formulas, x is a variable, and u and t are terms.

$(\alpha \vee \beta)$ abbreviates $((\neg \alpha) \to \beta)$.
$(\alpha \wedge \beta)$ abbreviates $(\neg (\alpha \to (\neg \beta)))$.
$(\alpha \leftrightarrow \beta)$ abbreviates $((\alpha \to \beta) \wedge (\beta \to \alpha))$; i.e.,

$$(\neg ((\alpha \to \beta) \to (\neg (\beta \to \alpha)))).$$

$\exists x\, \alpha$ abbreviates $(\neg \forall x (\neg \alpha))$.
$u = t$ abbreviates $=ut$. A similar abbreviation applies to some other two-place predicate and function symbols. For example, $2 < 3$ abbreviates $<2\,3$ and $2 + 2$ abbreviates $+2\,2$.
$u \neq t$ abbreviates $(\neg =ut)$; similarly $u \not< t$ abbreviates $(\neg <ut)$.

For parentheses we will use not only (and) but also [and], etc. And we omit mention of just as many as we possibly can. Toward that end we adopt the following conventions:

1. Outermost parentheses may be dropped. For example, $\forall x\, \alpha \to \beta$ is $(\forall x\, \alpha \to \beta)$.
2. $\neg$, $\forall$, and $\exists$ apply to as little as possible. For example,

$$\neg \alpha \wedge \beta \ \text{ is } \ ((\neg \alpha) \wedge \beta), \quad \text{and not} \quad \neg (\alpha \wedge \beta);$$
$$\forall x\, \alpha \to \beta \ \text{ is } \ (\forall x\, \alpha \to \beta), \quad \text{and not} \quad \forall x (\alpha \to \beta);$$
$$\exists x\, \alpha \wedge \beta \ \text{ is } \ (\exists x\, \alpha \wedge \beta), \quad \text{and not} \quad \exists x (\alpha \wedge \beta).$$

In such cases we might even add gratuitous parentheses, as in $(\exists x\, \alpha) \wedge \beta$.
3. $\wedge$ and $\vee$ apply to as little as possible, subject to item 2. For example,

$$\neg \alpha \wedge \beta \to \gamma \ \text{ is } \ ((\neg \alpha) \wedge \beta) \to \gamma.$$

4. When one connective is used repeatedly, the expression is grouped to the right. For example,

$$\alpha \to \beta \to \gamma \ \text{ is } \ \alpha \to (\beta \to \gamma).$$

EXAMPLES of how we can eliminate abbreviations, rewriting the formula in an unabbreviated way that explicitly lists each symbol in order:
1. $\exists x (Ax \wedge Bx)$ is $(\neg \forall x (\neg (\neg (Ax \to (\neg Bx)))))$.
But $(\neg \forall x (Ax \to (\neg Bx)))$ would be an equivalent formula (in any reasonable notion of equivalence).

2. $\exists x\, Ax \to Bx$ is $((\neg \forall x (\neg Ax)) \to Bx)$.
$\exists x (Ax \to Bx)$ is $(\neg \forall x (\neg (Ax \to Bx)))$.

We will try to use the various alphabets in a systematic way. The system is listed below, but there will be occasional exceptions for special reasons.

Predicate symbols: Uppercase italic letters. Also $\in$, $<$.

Variables: v_i, u, v, x, y, z.

Function symbols: f, g, h. Also $\mathbf{S}$, $+$, etc.

Constant symbols: a, b, $\ldots$. Also $\mathbf{0}$.

Terms: u, t.

Formulas: Lowercase Greek letters.

Sentences: σ, τ.

Sets of formulas: Uppercase Greek letters, plus certain italic letters that pretend to be Greek, viz., A (alpha) and T (tau).

Structures (see the next section): Uppercase German (Fraktur) letters.

Exercises

1. Assume that we have a language with the following parameters: $\forall$, intended to mean "for all things"; N, intended to mean "is a number"; I, intended to mean "is interesting"; $<$, intended to mean "is less than"; and $\mathbf{0}$, a constant symbol intended to denote zero. Translate into this language the English sentences listed below. If the English sentence is ambiguous, you will need more than one translation.

   (a) Zero is less than any number.

   (b) If any number is interesting, then zero is interesting.

   (c) No number is less than zero.

   (d) Any uninteresting number with the property that all smaller numbers are interesting certainly is interesting.

   (e) There is no number such that all numbers are less than it.

   (f) There is no number such that no number is less than it.

2. With the same language as in the preceding exercise, translate back into good English the wff

$$\forall x(Nx \to Ix \to \neg \forall y(Ny \to Iy \to \neg x < y)).$$

   In 3–8, translate each English sentence into the first-order language specified. (You may want to carry out the translation in several steps, as in some of the examples.) Make full use of the notational conventions and abbreviations to make the end result as readable as possible.

3. Neither a nor b is a member of every set. ($\forall$, for all sets; $\in$, is a member of; a, a; b, b.)

4. If horses are animals, then heads of horses are heads of animals. ($\forall$, for all things; E, is a horse; A, is an animal; hx, the head of x.)

5. (a) You can fool some of the people all of the time. (b) You can fool all of the people some of the time. (c) You can't fool all of the people all of the time. ($\forall$, for all things; P, is a person; T, is a

time; Fxy, you can fool x at y. One or more of the above may be ambiguous, in which case you will need more than one translation.)

6. (a) Adams can't do every job right. (b) Adams can't do any job right. ($\forall$, for all things; J, is a job; a, Adams; Dxy, x can do y right.)

7. (a) Nobody likes everybody. (b) No Democrat likes every Republican. ($\forall$, for all people; Lxy, x likes y; D, is a Democrat; R, is a Republican.)

8. (a) Every farmer who owns a donkey needs hay. (b) Every farmer who owns a donkey beats it. ($\forall$, for all things; F, is a farmer; D, is a donkey; Oxy, x owns y; H, needs hay; Bxy, x beats y.)

9. Give a precise definition of what it means for the variable x to occur free as the ith symbol in the wff α. (If x is the ith symbol of α but does not occur free there, then it is said to occur *bound* there.)

10. Rewrite each of the following wffs in a way which explicitly lists each symbol in the actual order:
    (a) $\exists\, v_1 P v_1 \land P v_1$.
    (b) $\forall\, v_1 A v_1 \land B v_1 \rightarrow \exists\, v_2 \neg C v_2 \lor D v_2$.
    In each case, say which variables occur free in the wff.

SECTION 2.2
Truth and Models

In sentential logic we had truth assignments to tell us which sentence symbols were to be interpreted as being true and which as false. In first-order logic the analogous role is played by *structures*, which can be thought of as providing the dictionary for translations from the formal language into English. (Structures are sometimes called *interpretations*, but we prefer to reserve that word for another concept, to be encountered in Section 2.7.)

A structure for a first-order language will tell us

1. What collection of things the universal quantifier symbol ($\forall$) refers to, and

2. What the other parameters (the predicate and function symbols) denote.

Formally, a *structure* $\mathfrak{A}$ for our given first-order language is a function whose domain is the set of parameters and such that[1]

---

[1] The symbol "$\mathfrak{A}$" is the letter A in the German (Fraktur) alphabet. The next two letters are $\mathfrak{B}$ and $\mathfrak{C}$.

1. $\mathfrak{A}$ assigns to the quantifier symbol $\forall$ a nonempty set $|\mathfrak{A}|$ called the *universe* (or *domain*) of $\mathfrak{A}$.

2. $\mathfrak{A}$ assigns to each n-place predicate symbol P an n-ary relation $P^{\mathfrak{A}} \subseteq |\mathfrak{A}|^n$; i.e., $P^{\mathfrak{A}}$ is a set of n-tuples of members of the universe.

3. $\mathfrak{A}$ assigns to each constant symbol c a member $c^{\mathfrak{A}}$ of the universe $|\mathfrak{A}|$.

4. $\mathfrak{A}$ assigns to each n-place function symbol f an n-ary operation $f^{\mathfrak{A}}$ on $|\mathfrak{A}|$; i.e., $f^{\mathfrak{A}} : |\mathfrak{A}|^n \rightarrow |\mathfrak{A}|$.

The idea is that $\mathfrak{A}$ assigns meaning to the parameters. $\forall$ is to mean "for everything in $|\mathfrak{A}|$." The symbol c is to name the point $c^{\mathfrak{A}}$. The atomic formula $P t_1 \cdots t_n$ is to mean that the n-tuple of points named by $t_1, \ldots, t_n$ is in the relation $P^{\mathfrak{A}}$. (We will shortly restate these conditions more carefully.)

Note that we require the universe $|\mathfrak{A}|$ to be nonempty. Notice also that $f^{\mathfrak{A}}$ must have all of $|\mathfrak{A}|^n$ in for its domain; we have made no provision for partially defined functions.

> **EXAMPLE.**   Consider the language for set theory, whose only parameter (other than $\forall$) is $\in$. Take the structure $\mathfrak{A}$ with
>
> $$|\mathfrak{A}| = \text{the set of natural numbers,}$$
> $$\in^{\mathfrak{A}} = \text{the set of pairs } \langle m, n \rangle \text{ such that } m < n.$$
>
> (Thus we translate $\in$ as "is less than.") In the presence of a structure we can translate sentences from the formal language into English and attempt to say whether these translations are true or false. The sentence of this first-order language
>
> $$\exists x \, \forall y \, \neg \, y \in x$$
>
> (or more formally, $(\neg \forall v_1(\neg \forall v_2(\neg \in v_2 v_1)))$), which under another translation asserts the existence of an empty set, is now translated under $\mathfrak{A}$ into
>
> > There is a natural number such that no natural number is smaller,
>
> which is true. Because of this we will say that $\exists x \, \forall y \, \neg \, y \in x$ is *true* in $\mathfrak{A}$, or that $\mathfrak{A}$ is a *model* of the sentence. On the other hand, $\mathfrak{A}$ is not a model of the pair-set axiom,
>
> $$\forall x \, \forall y \, \exists z \, \forall t (t \in z \leftrightarrow t = x \lor t = y),$$
>
> as the translation of this sentence under $\mathfrak{A}$ is false. For there is no natural number m such that for every n,
>
> $$n < m \quad \text{iff} \quad n = 1.$$
>
> (The reader familiar with axiomatic set theory can check that $\mathfrak{A}$

is a model of the extensionality axiom, the union axiom, and the axiom of regularity.)

EXAMPLE.   Again assume the language has only the parameters $\forall$ and a two-place predicate symbol E. But this time, consider the *finite* structure $\mathfrak{B}$ with universe $|\mathfrak{B}|$ consisting of a set of four distinct objects $\{a, b, c, d\}$. Suppose the binary relation $E^{\mathfrak{B}}$ is the following set of pairs:

$$E^{\mathfrak{B}} = \{\langle a, b\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, c\rangle\}.$$

Then we can picture $\mathfrak{B}$ as the *directed graph* whose vertex-set is the universe $\{a, b, c, d\}$:
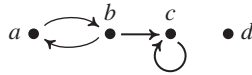
Here we interpret Exy as saying there is an *edge* from vertex x to vertex y in the graph. (If the binary relation $E^{\mathfrak{B}}$ had been symmetric, then we could have pictured the structure as an undirected graph.)

Consider now the sentence $\exists x \forall y \neg yEx$. Under the structure $\mathfrak{B}$, we can translate this as follows:

There is a vertex such that for any vertex,
no edge points from the latter to the former.

(The English version is harder to read than the symbolic one!) This sentence is true in $\mathfrak{B}$, because no edge points to the vertex d.

In the preceding examples it was intuitively pretty clear that certain sentences of the formal language were true in the structure and some were false. But we want a precise mathematical definition of "σ is true in $\mathfrak{A}$." This should be stated in mathematical terms, without employing translations into English or a supposed criterion for asserting that some English sentences are true while the others are false. (If you think you have such a criterion, try it on the sentence "This sentence is false.") In other words, we want to take our informal concept of "σ is true in $\mathfrak{A}$" and make it part of *mathematics*.

In order to define "σ is true in $\mathfrak{A}$,"

$$\models_{\mathfrak{A}} \sigma,$$

for sentences σ and structures $\mathfrak{A}$, we will find it desirable first to define a more general concept involving wffs. Let

φ be a wff of our language,
$\mathfrak{A}$ a structure for the language,

$s : V \to |\mathfrak{A}|$ a function from the set V of all variables into the universe $|\mathfrak{A}|$ of $\mathfrak{A}$.

Then we will define what it means for $\mathfrak{A}$ to *satisfy* φ *with* s,

$$\models_{\mathfrak{A}} \varphi[s].$$

The informal version is

$\models_{\mathfrak{A}} \varphi[s]$ if and only if the translation of φ determined by $\mathfrak{A}$, where the variable x is translated as $s(x)$ wherever it occurs free, is true.

The formal definition of satisfaction proceeds as follows:
I. *Terms*. We define the extension

$$\bar{s} : T \to |\mathfrak{A}|,$$

a function from the set T of all terms into the universe of $\mathfrak{A}$. The idea is that $\bar{s}(t)$ should be the member of the universe $|\mathfrak{A}|$ that is named by the term t. $\bar{s}$ is defined by recursion as follows:

1. For each variable x, $\bar{s}(x) = s(x)$.
2. For each constant symbol c, $\bar{s}(c) = c^{\mathfrak{A}}$.
3. If $t_1, \ldots, t_n$ are terms and f is an n-place function symbol, then

$$\bar{s}(f t_1 \cdots t_n) = f^{\mathfrak{A}}(\bar{s}(t_1), \ldots, \bar{s}(t_n)).$$

A commutative diagram, for $n = 1$, is

The existence of a unique such extension $\bar{s}$ of s follows from the recursion theorem (Section 1.4), by using the fact that the terms have unique decompositions (Section 2.3). Notice that $\bar{s}$ depends both on s and on $\mathfrak{A}$. (In fact a reasonable alternative notation for $\bar{s}(t)$ would be $t^{\mathfrak{A}}[s]$, which explicitly displays the dependence on $\mathfrak{A}$.)

II. *Atomic formulas*. The atomic formulas were defined explicitly, not inductively. The definition of satisfaction of atomic formulas is therefore also explicit, and not recursive.

1. $\models_{\mathfrak{A}} = t_1 t_2[s]$ iff $\bar{s}(t_1) = \bar{s}(t_2)$.
(Thus $=$ means $=$. Note that $=$ is a logical symbol, not a parameter open to interpretation.)
2. For an n-place predicate parameter P,

$$\models_{\mathfrak{A}} P t_1 \cdots t_n[s] \text{ iff } \langle \bar{s}(t_1), \ldots, \bar{s}(t_n) \rangle \in P^{\mathfrak{A}}.$$

III. *Other wffs*. The wffs we defined inductively, and consequently here satisfaction is defined recursively.

1. For atomic formulas, the definition is above.
2. $\models_{\mathfrak{A}} \neg \varphi[s]$ iff $\not\models_{\mathfrak{A}} \varphi[s]$.
3. $\models_{\mathfrak{A}} (\varphi \rightarrow \psi)[s]$ iff either $\not\models_{\mathfrak{A}} \varphi[s]$ or $\models_{\mathfrak{A}} \psi[s]$ or both.

(In other words, if $\mathfrak{A}$ satisfies φ with s then $\mathfrak{A}$ satisfies ψ with s.)

4. $\models_{\mathfrak{A}} \forall x\, \varphi[s]$ iff for every $d \in |\mathfrak{A}|$, we have $\models_{\mathfrak{A}} \varphi[s(x \mid d)]$.

Here $s(x \mid d)$ is the function which is exactly like s except for one thing: At the variable x it assumes the value d. This can be expressed by the equation

$$s(x \mid d)(y) = \begin{cases} s(y) & \text{if } y \neq x, \\ d & \text{if } y = x. \end{cases}$$

(Thus $\forall$ means "for all things in $|\mathfrak{A}|$.")

At this point the reader might want to reconsider the informal version of $\models_{\mathfrak{A}} \varphi[s]$ on page 83 and observe how it has been formalized.

We should remark that the definition of satisfaction is another application of the recursion theorem together with the fact that the wffs have unique decompositions. The definition can be restated in terms of functions to make it clearer how the recursion theorem of Section 1.4 applies:

(i) Consider one fixed $\mathfrak{A}$.

(ii) Define a function $\bar{h}$ (extending a function h defined on atomic formulas) such that for any wff φ, $\bar{h}(\varphi)$ is a set of functions from V into $|\mathfrak{A}|$.

(iii) Define

$$\models_{\mathfrak{A}} \varphi[s] \quad \text{iff} \quad s \in \bar{h}(\varphi).$$

We leave to the reader the exercise of writing down the explicit definition of h and the clauses that uniquely determine its extension $\bar{h}$. (See Exercise 7.) An elegant alternative is to have $\bar{h}(\varphi)$ be a set of functions on the set of those variables that occur free in φ.

EXAMPLE. Assume that our language has the parameters $\forall$, P (a two-place predicate symbol), f (a one-place function symbol), and c (a constant symbol). Let $\mathfrak{A}$ be the structure for his language defined as follows:

$|\mathfrak{A}| = \mathbb{N}$, the set of all natural numbers,
$P^{\mathfrak{A}} = $ the set of pairs $\langle m, n \rangle$ such that $m \leq n$,
$f^{\mathfrak{A}} = $ the successor function S; $f^{\mathfrak{A}}(n) = n + 1$,
$c^{\mathfrak{A}} = 0$.

We can summarize this in one line, by suppressing the fact that $\mathfrak{A}$

is really a function and merely listing its components:

$$\mathfrak{A} = (\mathbb{N}; \leq, S, 0).$$

This notation is unambiguous only when the context makes clear just which components go with which parameters.

Let $s : V \to \mathbb{N}$ be the function for which $s(v_i) = i - 1$; i.e., $s(v_1) = 0$, $s(v_2) = 1$, and so forth.

   1. $\bar{s}(ff v_3) = S(S(2)) = 4$ and $\bar{s}(f v_1) = S(0) = 1$.
   2. $\bar{s}(c) = 0$ and $\bar{s}(ffc) = 2$; no use is made of s.
   3. $\models_{\mathfrak{A}} Pcf v_1[s]$. This is informally obvious, since when we translate back into English we get the true sentence "$0 \leq 1$." More formally, the reason is that

$$\langle \bar{s}(c), \bar{s}(f v_1) \rangle = \langle 0, 1 \rangle \in P^{\mathfrak{A}}.$$

   4. $\models_{\mathfrak{A}} \forall v_1 \, Pcv_1$. The translation into English is "0 is less than or equal to any natural number." Formally we must verify that for any n in $\mathbb{N}$,

$$\models_{\mathfrak{A}} Pcv_1[s(v_1 \mid n)],$$

which reduces to

$$\langle 0, n \rangle \in P^{\mathfrak{A}},$$

i.e., $0 \leq n$.
   5. $\not\models_{\mathfrak{A}} \forall v_1 \, P v_2 v_1[s]$ because there is a natural number m such that

$$\not\models_{\mathfrak{A}} P v_2 v_1[s(v_1 \mid m)];$$

i.e.,

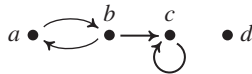$$\langle s(v_2), m \rangle \notin P^{\mathfrak{A}}.$$

In fact, since $s(v_2) = 1$, we must take m to be 0.

   The reader is to be cautioned against confusing, for example, the function *symbol* f with the *function* $f^{\mathfrak{A}}$.

EXAMPLE.   Previously we considered the structure $\mathfrak{B}$ with

$$|\mathfrak{B}| = \{a, b, c, d\} \quad \text{and} \quad E^{\mathfrak{B}} = \{\langle a, b \rangle, \langle b, a \rangle, \langle b, c \rangle, \langle c, c \rangle\}$$

for the language with parameters $\forall$ and E:

Then $\models_{\mathfrak{B}} \forall v_2 \neg E v_2 v_1[s]$ iff $s(v_1) = d$. That is, there is no edge pointing to the vertex d, but d is the only such vertex. Taking the negation of the formula, we have $\models_{\mathfrak{B}} \exists v_2 E v_2 v_1[s]$ iff $s(v_1) \in \{a, b, c\}$.

At this point we pause to verify that when we want to know whether or not a structure $\mathfrak{A}$ satisfies a wff φ with s, we do not really need all of the (infinite amount of) information s gives us. All that matter are the values of the function s at the (finitely many) variables which occur free in φ. In particular, if φ is a sentence, then s does not matter at all.

THEOREM 22A   Assume that s_1 and s_2 are functions from V into $|\mathfrak{A}|$ which agree at all variables (if any) that occur free in the wff φ. Then

$$\models_{\mathfrak{A}} \varphi[s_1] \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi[s_2].$$

PROOF.   Because satisfaction was defined recursively, this proof uses induction. We consider the fixed structure $\mathfrak{A}$ and show by induction that every wff φ has the property that whenever two functions s_1, s_2 agree on the variables free in φ then $\mathfrak{A}$ satisfies φ with s_1 iff it does so with s_2.

Case 1: $\varphi = P t_1 \cdots t_n$ is atomic. Then any variable in φ occurs free. Thus s_1 and s_2 agree at all the variables in each t_i. It follows that $\bar{s}_1(t_i) = \bar{s}_2(t_i)$ for each i; a detailed proof would use induction on t_i. Consequently, $\mathfrak{A}$ satisfies $P t_1 \cdots t_n$ with s_1 iff it does so with s_2.

Cases 2 and 3: φ has the form $\neg \alpha$ or $\alpha \rightarrow \beta$. These cases are immediate from the inductive hypothesis.

Case 4: $\varphi = \forall x \, \psi$. Then the variables free in φ are those free in ψ with the exception of x. Thus for any d in $|\mathfrak{A}|$, $s_1(x \mid d)$ and $s_2(x \mid d)$ agree at all variables free in ψ. By inductive hypothesis, then, $\mathfrak{A}$ satisfies ψ with $s_1(x \mid d)$ iff it does so with $s_2(x \mid d)$. From this and the definition of satisfaction we see that $\mathfrak{A}$ satisfies $\forall x \, \psi$ with s_1 iff it does so with s_2.   ⊣

In effect, the above proof amounts to looking through the definition of satisfaction and seeing what information given by s was actually used. There is an analogous fact regarding structures: If $\mathfrak{A}$ and $\mathfrak{B}$ agree at all the parameters that occur in φ, then $\models_{\mathfrak{A}} \varphi[s]$ iff $\models_{\mathfrak{B}} \varphi[s]$.

This theorem justifies the following notation: Suppose that φ is a formula such that all variables occurring free in φ are included among $v_1, \ldots, v_k$. Then for elements $a_1, \ldots, a_k$ of $|\mathfrak{A}|$,

$$\models_{\mathfrak{A}} \varphi[\![a_1, \ldots, a_k]\!]$$

means that $\mathfrak{A}$ satisfies φ with some (and hence with any) function $s : V \to |\mathfrak{A}|$ for which $s(v_i) = a_i$, $1 \leq i \leq k$. To return to a recent example where $\mathfrak{A} = (\mathbb{N}; \leq, S, 0)$, we have $\models_{\mathfrak{A}} \forall v_2\, P v_1 v_2 [\![0]\!]$ but $\not\models_{\mathfrak{A}} \forall v_2\, P v_1 v_2 [\![5]\!]$.

COROLLARY 22B   For a sentence σ, either

    (a) $\mathfrak{A}$ satisfies σ with every function s from V into $|\mathfrak{A}|$, or
    (b) $\mathfrak{A}$ does not satisfy σ with any such function.

If alternative (a) holds, then we say that σ is *true* in $\mathfrak{A}$ (written $\models_{\mathfrak{A}} \sigma$) or that $\mathfrak{A}$ is a *model* of σ. And if alternative (b) holds, then of course σ is *false* in $\mathfrak{A}$. (They cannot both hold since $|\mathfrak{A}|$ is nonempty). $\mathfrak{A}$ is a *model* of a set Σ of sentences iff it is a model of every member of Σ.

EXAMPLE.   If $\mathfrak{R}$ is the real field, $(\mathbb{R}; 0, 1, +, \times)$, and $\mathfrak{Q}$ is the rational field, $(\mathbb{Q}; 0, 1, +, \times)$, is there a sentence true in one and false in the other? Yes; because $\sqrt{2}$ is irrational, the sentence $\exists x (x \cdot x = 1 + 1)$ is false in the rational field, but true in the real field.

EXAMPLE.   Suppose our given language has only the parameters $\forall$ and P, where P is a two-place predicate symbol. Then a structure $\mathfrak{A}$ is determined by the universe $|\mathfrak{A}|$ and the binary relation $P^{\mathfrak{A}}$. With some minor illegality we again write

$$\mathfrak{A} = (|\mathfrak{A}|; P^{\mathfrak{A}}).$$

Now consider the problem of characterizing the class of all models of the following sentences:

    1. $\forall x \forall y\, x = y$. A structure $(A; R)$ is a model of this iff A contains exactly one element. R can either be empty or can be the singleton $A \times A$.
    2. $\forall x \forall y\, Pxy$. A structure $(A; R)$ is a model of this iff $R = A \times A$. A can be any nonempty set.
    3. $\forall x \forall y \neg Pxy$. A structure $(A; R)$ is a model of this iff $R = \varnothing$.
    4. $\forall x \exists y\, Pxy$. The condition for $(A; R)$ to be a model of this is that the domain of R is A.

The notational conventions adopted earlier were done in a rational way:

    1. $\models_{\mathfrak{A}} (\alpha \wedge \beta)[s]$ iff $\models_{\mathfrak{A}} \alpha[s]$ and $\models_{\mathfrak{A}} \beta[s]$; similarly for $\vee$ and $\leftrightarrow$.
    2. $\models_{\mathfrak{A}} \exists x\, \alpha[s]$ iff there is some $d \in |\mathfrak{A}|$ with the property that $\models_{\mathfrak{A}} \alpha[s(x \,|\, d)]$.

The proof for the second of these is as follows:

$$\models_{\mathfrak{A}} \exists x\, \alpha[s] \quad \text{iff} \quad \models_{\mathfrak{A}} \neg \forall x\, \neg \alpha[s],$$

iff $\quad \not\models_{\mathfrak{A}} \forall x\, \neg \alpha[s],$

iff it is not the case that for all d in $|\mathfrak{A}|$,

$$\models_{\mathfrak{A}} \neg \alpha[s(x \mid d)],$$

iff it is not the case that for all d in $|\mathfrak{A}|$,

$$\not\models_{\mathfrak{A}} \alpha[s(x \mid d)],$$

iff for some d in $|\mathfrak{A}|$, $\models_{\mathfrak{A}} \alpha[s(x \mid d)].$

Logical Implication

We now have the necessary tools to formulate the important concept of logical implication for our language.

DEFINITION.   Let Γ be a set of wffs, φ a wff. Then Γ *logically implies* φ, written $\Gamma \models \varphi$, iff for every structure $\mathfrak{A}$ for the language and every function $s : V \to |\mathfrak{A}|$ such that $\mathfrak{A}$ satisfies every member of Γ with s, $\mathfrak{A}$ also satisfies φ with s.

We use the same symbol, "$\models$," that was used in Chapter 1 for tautological implication. But henceforth it will be used solely for logical implication. As before we will write "$\gamma \models \varphi$" in place of "$\{\gamma\} \models \varphi$." Say that φ and ψ are *logically equivalent* ($\varphi \models\models \psi$) iff $\varphi \models \psi$ and $\psi \models \varphi$.

The first-order analogue of the concept of a tautology is the concept of a valid formula: A wff φ is *valid* iff $\varnothing \models \varphi$ (written simply "$\models \varphi$"). Thus φ is valid iff for every $\mathfrak{A}$ and every $s : V \to |\mathfrak{A}|$, $\mathfrak{A}$ satisfies φ with s.

For sentences, the concept of logical implication can be stated more concisely, by applying Theorem 22A:

COROLLARY 22C   For a set Σ; τ of sentences, $\Sigma \models \tau$ iff every model of Σ is also a model of τ. A sentence τ is valid iff it is true in every structure.

EXAMPLES   of logical implication. Readers are invited to convince themselves of each of the following:

1. $\forall v_1\, Q v_1 \models Q v_2.$
2. $Q v_1 \not\models \forall v_1\, Q v_1.$ Here it suffices to find just one structure $\mathfrak{A}$ and just one function $s : V \to |\mathfrak{A}|$ such that on the one hand, $\models_{\mathfrak{A}} Q v_1[s]$ but, on the other hand, $\mathfrak{A}$ is not a model of $\forall v_1\, Q v_1$. $|\mathfrak{A}|$ will need to have at least two members.
3. $\models \neg\neg\sigma \to \sigma.$ If $\mathfrak{A}$ is a model of $\neg\neg\sigma$, then $\not\models_{\mathfrak{A}} \neg\sigma$ whence $\models_{\mathfrak{A}} \sigma$. But one might raise the objection: Aren't we

using here the law of double negation, the law we are claiming to prove? The answer is a definite yes and no. We are proving the law of double negation for the formal language we are talking about (sometimes called the *object language*). In doing so, we can of course use any correct reasoning (out in the meta-language, English), exactly as we might do in reasoning about vector spaces or graphs. In particular, the reasoning might involve principles that when formally *modeled* would involve $\neg\neg\sigma$ and σ. There is no circularity. But the meta-language sentences we use are — unsurprisingly — related to the object language formulas we talk about. In this connection, see the book's only picture (at the end of Section 2.4).

    4. $\forall v_1\, Q v_1 \models \exists v_2\, Q v_2$.   Recall that the universe of any structure is nonempty.

    5. $\exists x \forall y\, Pxy \models \forall y \exists x\, Pxy$.   This example will come up again in Section 2.4.

    6. $\forall y \exists x\, Pxy \not\models \exists x \forall y\, Pxy$.

    7. $\models \exists x (Qx \rightarrow \forall x\, Qx)$.   This is a strange — but valid — sentence.

The definition of logical implication is very much like the definition of tautological implication in Chapter 1. But there is an important difference of complexity. Suppose in sentential logic you want to know whether or not a wff α is a tautology. The definition requires that you consider finitely many truth assignments, each of which is a finite function. For each such truth assignment v, you must calculate $\bar{v}(\alpha)$, which can be effectively done in a finite length of time. (Consequently, the set of tautologies is decidable, as was observed before.)

In contrast to the finitary procedure for tautologies, suppose that you want to know whether or not a wff φ (of our first-order language) is valid. The definition requires that you consider every structure $\mathfrak{A}$. (In particular this requires using every nonempty set, of which there are a great many.) For each of these structures, you then must consider each function s from the set V of variables into $|\mathfrak{A}|$. And for each given $\mathfrak{A}$ and s, you must determine whether or not $\mathfrak{A}$ satisfies φ with s. When $|\mathfrak{A}|$ is infinite, this is a complicated notion in itself.

In view of these complications, it is not surprising that the set of valid formulas fails to be decidable (cf. Section 3.5). What is surprising is that the concept of validity turns out to be equivalent to another concept (deducibility) whose definition is much closer to being finitary. (See Section 2.4.) Using that equivalence we will be able to show (under some reasonable assumptions) that the set of validities (i.e., the set of valid wffs) is effectively enumerable. The effective enumeration procedure yields a more concrete characterization of the set of valid formulas.

Definability in a Structure

Suppose we want to study the real field, $(\mathbb{R}; 0, 1, +, \cdot)$, consisting of the set $\mathbb{R}$ of real numbers, together with the distinguished elements 0 and 1 and the two operations of addition and multiplication. We can consider the real field as a structure

$$\mathfrak{R} = (\mathbb{R}; 0, 1, +, \cdot)$$

where the language (with equality) has constant symbols $\mathbf{0}$ and $\mathbf{1}$ and two-place function symbols $+$ and $\cdot$.

Although we have not included an ordering symbol $<$ in the language, we still have a way to say "$x \geq 0$." Because *in this structure*, the nonnegative elements are exactly the elements with square roots. That is, the formula $\exists\, v_2\ x = v_2 \cdot v_2$ is satisfied in the structure $\mathfrak{R}$ whenever x is assigned a nonnegative number, and only then:

$$\models_{\mathfrak{R}} \exists\, v_2\ v_1 = v_2 \cdot v_2\ [\![a]\!] \iff a \geq 0.$$

Because of this fact, we will say that the interval $[0, \infty)$ is *definable* in $\mathfrak{R}$, and that the formula $\exists\, v_2\ v_1 = v_2 \cdot v_2$ defines it.

Moreover, the ordering relation on the reals, i.e., the binary relation

$$\{\langle a, b \rangle \in \mathbb{R} \times \mathbb{R} \mid a \leq b\},$$

is defined in the structure $\mathfrak{R}$ by the formula expressing "$v_1 \leq v_2$":

$$\exists v_3\ v_2 = v_1 + v_3 \cdot v_3.$$

For a smaller example, take the directed graph

$$\mathfrak{A} = (\{a, b, c\}; \{\langle a, b \rangle, \langle a, c \rangle\})$$

where the language has parameters $\forall$ and E:

$$b \bullet \leftarrow \overset{a}{\bullet} \rightarrow \bullet\, c$$

Then in $\mathfrak{A}$, the set $\{b, c\}$ (the range of the relation $E^{\mathfrak{A}}$) is defined by the formula $\exists\, v_2\ E v_2 v_1$. In contrast, the set $\{b\}$ is *not* definable in $\mathfrak{A}$. This is because there is no definable property in this structure that would separate b and c; the proof of this fact will utilize the homomorphism theorem, to be proved later in this section.

We now want to set forth precisely this concept of definability of a subset of the universe or of a relation on the universe. Consider a structure $\mathfrak{A}$ and a formula φ whose free variables are among $v_1, \ldots, v_k$. Then we can construct the k-ary relation on $|\mathfrak{A}|$

$$\{\langle a_1, \ldots, a_k \rangle \mid \models_{\mathfrak{A}} \varphi[\![a_1, \ldots, a_k]\!]\}.$$

Call this the k-ary relation φ *defines* in $\mathfrak{A}$. In general, a k-ary relation

on $|\mathfrak{A}|$ is said to be *definable* in $\mathfrak{A}$ iff there is a formula (whose free variables are among $v_1, \ldots, v_k$) that defines it there.

EXAMPLE.   Assume that we have a part of the language for number theory, specifically that our language has the parameters $\forall$, $\mathbf{0}$, $\mathbf{S}$, $+$, and $\cdot$. Let $\mathfrak{N}$ be the intended structure:

$|\mathfrak{N}| = \mathbb{N}$, the set of natural numbers.
$\mathbf{0}^{\mathfrak{N}} = 0$, the number 0.
$\mathbf{S}^{\mathfrak{N}}$, $+^{\mathfrak{N}}$, and $\cdot^{\mathfrak{N}}$ are S, $+$, and $\cdot$, the functions of successor, addition, and multiplication.

In one equation,

$$\mathfrak{N} = (\mathbb{N}; 0, S, +, \cdot).$$

Some relations on $\mathbb{N}$ are definable in $\mathfrak{N}$ and some are not. One way to show that some are not definable is to use the fact that there are uncountably many relations on $\mathbb{N}$ but only countably many possible defining formulas. (There is, however, an inherent difficulty in giving a specific example. After all, if something is undefinable, then it is hard to say exactly what it is! Later we will get to see a specific example, the set of Gödel numbers of sentences true in $\mathfrak{N}$; see Section 3.5.)

1. The ordering relation $\{\langle m, n \rangle \mid m < n\}$ is defined in $\mathfrak{N}$ by the formula

$$\exists v_3 \; v_1 + \mathbf{S}v_3 = v_2.$$

2. For any natural number n, $\{n\}$ is definable. For example, $\{2\}$ is defined by the equation

$$v_1 = \mathbf{SS0}.$$

Because of this we say that n is a *definable element* in $\mathfrak{N}$.

3. The set of primes is definable in $\mathfrak{N}$. We could use the formula

$$\mathbf{1} < v_1 \wedge \forall v_2 \, \forall v_3 (v_1 = v_2 \cdot v_3 \rightarrow v_2 = \mathbf{1} \vee v_3 = \mathbf{1})$$

if we had parameters $\mathbf{1}$ and $<$ for 1 and $<$. But since $\{1\}$ and $<$ are definable in $\mathfrak{N}$, it is really quite unnecessary to add parameters for them; we can simply use their definitions instead. Thus the set of primes is definable by

$$\exists v_3 \, \mathbf{S0} + \mathbf{S}v_3 = v_1 \wedge \forall v_2 \, \forall v_3 \; (v_1 = v_2 \cdot v_3 \rightarrow$$
$$v_2 = \mathbf{S0} \vee v_3 = \mathbf{S0}).$$

4. Exponentiation, $\{\langle m, n, p \rangle \mid p = m^n\}$ is also definable in $\mathfrak{N}$. This is by no means obvious; we will give a proof later (in Section 3.8) using the Chinese remainder theorem.

In fact, we will argue later that any decidable relation on $\mathbb{N}$ is definable in $\mathfrak{N}$, as is any effectively enumerable relation and a great many others. To some extent the complexity of a definable relation can be measured by the complexity of the simplest defining formula. This idea will come up again at the end of Section 3.5.

Definability of a Class of Structures

Many a mathematics class, on its first day, begins with the instructor saying something like one of the following:

1. "A *graph* is defined to consist of a nonempty set V together with a set E such that. ..."
2. "A *group* is defined to consist of a nonempty set G together with a binary operation $\circ$ satisfying the axioms. ..."
3. "An *ordered field* is defined to consist of a nonempty set F together with two binary operations $+$ and $\cdot$ and a binary relation $<$ satisfying the axioms. ..."
4. "A *vector space* is defined to consist of a nonempty set V together with a binary operation $+$ and, for each real number r, an operation called scalar multiplication such that. ..."

We want to abstract from this situation. In each case, the objects of study (the graphs, the groups, and so forth) are *structures* for a suitable language. Moreover, they are required to satisfy a certain set Σ of sentences (referred to as "axioms"). The course in question then studies the models of the set Σ of axioms — or at least some of the models.

For a set Σ of sentences, let Mod Σ be the class of all models of Σ, i.e., the class of all structures for the language in which every member of Σ is true. For a single sentence τ we write simply "Mod τ" instead of "Mod $\{\tau\}$." (The reader familiar with axiomatic set theory will notice that Mod Σ, if nonempty, is a proper class; i.e., it is too large to be a set.)

A class $\mathcal{K}$ of structures for our language is an *elementary class* (EC) iff $\mathcal{K} = $ Mod τ for some sentence τ. $\mathcal{K}$ is an *elementary class in the wider sense* (EC_Δ) iff $\mathcal{K} = $ Mod Σ for some set Σ of sentences. (The adjective "elementary" is employed as a synonym for "first-order.")

EXAMPLES

1. Assume that the language has equality and the two parameters $\forall$ and E, where E is a two-place predicate symbol. Then a *graph* is a structure for this language $\mathfrak{A} = (V; E^{\mathfrak{A}})$ consisting of a nonempty set V of objects called *vertices* (or *nodes*), and an *edge relation* $E^{\mathfrak{A}}$ that is symmetric (if $u E^{\mathfrak{A}} v$ then $v E^{\mathfrak{A}} u$) and irreflexive (never $v E^{\mathfrak{A}} v$). The axiom stating that the edge relation is symmetric and irreflexive can be translated by the sentence

$$\forall x(\neg\, x Ex \wedge \forall\, y(x Ey \rightarrow yEx)).$$

So the class of all graphs is an elementary class. For *directed graphs* or *digraphs*, the assumption of symmetry is dropped. And if one wants to allow "loops" then the assumption of irreflexivity is dropped. But perhaps the instructor then explains that the course will study only *finite* graphs. Is the class of all finite graphs an elementary class? No, we will prove later that it is not, not even in the wider sense.

2. Assume that the language has equality and the parameters $\forall$ and P, where P is a two-place predicate symbol. As before, a structure $(A; R)$ for the language consists of a nonempty set A together with a binary relation R on A. $(A; R)$ is called an *ordered set* iff R is transitive and satisfies the trichotomy condition (which states that for any a and b in A, exactly one of $\langle a, b\rangle \in R$, $a = b$, $\langle b, a\rangle \in R$ holds). Because these conditions can be translated into a sentence of the formal language, the class of nonempty ordered sets is an elementary class. It is, in fact, Mod τ, where τ is the conjunction of the three sentences

$$\forall x\, \forall y\, \forall z(x Py \rightarrow yPz \rightarrow x Pz);$$
$$\forall x\, \forall y(x Py \vee x = y \vee yPx);$$
$$\forall x\, \forall y(x Py \rightarrow \neg yPx).$$

The next two examples assume that the reader has had some contact with algebra.

3. Assume that the language has $=$ and the parameters $\forall$ and $\circ$, where $\circ$ is a two-place function symbol. The class of all groups is an elementary class, being the class of all models of the conjunction of the group axioms:

$$\forall x\, \forall y\, \forall z\, (x \circ y) \circ z = x \circ (y \circ z);$$
$$\forall x\, \forall y\, \exists z\, x \circ z = y;$$
$$\forall x\, \forall y\, \exists z\, z \circ x = y.$$

The class of all infinite groups is EC_Δ. To see this, let

$$\lambda_2 = \exists x\, \exists y\, x \neq y,$$
$$\lambda_3 = \exists x\, \exists y\, \exists z(x \neq y \wedge x \neq z \wedge y \neq z),$$
$$\ldots .$$

Thus λ_n translates, "There are at least n things." Then the group axioms together with $\{\lambda_2, \lambda_3, \ldots\}$ form a set Σ for which Mod Σ is exactly the class of infinite groups. We will eventually (in Section 2.6) be able to show that the class of infinite groups is not EC.

4. Assume that the language has equality and the parameters $\forall, \mathbf{0}, \mathbf{1}, +, \cdot$. Fields can be regarded as structures for this language.

The class of all fields is an elementary class. The class of fields of characteristic zero is EC_Δ. It is not EC, a fact which will follow from the compactness theorem for first-order logic (Section 2.6 again).

Homomorphisms[1]

In courses about graphs or groups or vector spaces, one usually encounters the concept of what it means for two of the structures in question, $\mathfrak{A}$ and $\mathfrak{B}$, to be *isomorphic*: Roughly speaking, there must be a one-to-one correspondence between their universes $|\mathfrak{A}|$ and $|\mathfrak{B}|$ that "preserves" the operations and relations.

It is then explained that two isomorphic structures, although not identical, must have all the same mathematical properties. We want to define here the isomorphism concept in a general setting, and to show that two isomorphic structures must satisfy exactly the same sentences.

Let $\mathfrak{A}$, $\mathfrak{B}$ be structures for the language. A *homomorphism* h of $\mathfrak{A}$ into $\mathfrak{B}$ is a function $h : |\mathfrak{A}| \rightarrow |\mathfrak{B}|$ with the properties:

(a) For each n-place predicate parameter P and each n-tuple $\langle a_1, \dots, a_n \rangle$ of elements of $|\mathfrak{A}|$,

$$\langle a_1, \dots, a_n \rangle \in P^{\mathfrak{A}} \quad \text{iff} \quad \langle h(a_1), \dots, h(a_n) \rangle \in P^{\mathfrak{B}}.$$

(b) For each n-place function symbol f and each such n-tuple,

$$h(f^{\mathfrak{A}}(a_1, \dots, a_n)) = f^{\mathfrak{B}}(h(a_1), \dots, h(a_n)).$$

In the case of a constant symbol c this becomes

$$h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}.$$

Conditions (a) and (b) are usually stated: "h preserves the relations and functions." (It must be admitted that some authors use a weakened version of condition (a); our homomorphisms are their "strong homomorphisms.")

If, in addition, h is one-to-one, it is then called an *isomorphism* (or an *isomorphic embedding*) of $\mathfrak{A}$ into $\mathfrak{B}$. If there is an isomorphism of $\mathfrak{A}$ onto $\mathfrak{B}$ (i.e., an isomorphism h for which $\operatorname{ran} h = |\mathfrak{B}|$), then $\mathfrak{A}$ and $\mathfrak{B}$ are said to be *isomorphic* (written $\mathfrak{A} \cong \mathfrak{B}$).

The reader has quite possibly encountered this concept before in special cases such as structures that are groups or fields.

---

[1] This topic can be postponed somewhat. But homomorphisms will be used in the proof of the completeness theorem (with equality). And we make use of the isomorphism concept, starting in Section 6 of Chapter 2.

EXAMPLE. Assume that we have a language with the parameters $\forall$, $+$, and $\cdot$. Let $\mathfrak{A}$ be the structure $(\mathbb{N}; +, \cdot)$. We can define a function $h : \mathbb{N} \to \{e, o\}$ by

$$h(n) = \begin{cases} e & \text{if } n \text{ is even,} \\ o & \text{if } n \text{ is odd.} \end{cases}$$

Then h is a homomorphism of $\mathfrak{A}$ onto $\mathfrak{B}$, where $|\mathfrak{B}| = \{e, o\}$ and $+^{\mathfrak{B}}, \cdot^{\mathfrak{B}}$ are given by the following tables:

$+^{\mathfrak{B}}$	e	o
e	e	o
o	o	e

$\cdot^{\mathfrak{B}}$	e	o
e	e	e
o	e	o

It can then be verified that condition (b) of the definition is satisfied. For example, if a and b are both odd numbers, then $h(a+b) = e$ and $h(a) +^{\mathfrak{B}} h(b) = o +^{\mathfrak{B}} o = e$.

EXAMPLE. Let $\mathbb{P}$ be the set of positive integers, let $<_P$ be the usual ordering relation on $\mathbb{P}$, and let $<_N$ be the usual ordering relation on $\mathbb{N}$. Then there is an isomorphism h from the structure $(\mathbb{P}; <_P)$ onto $(\mathbb{N}; <_N)$; we take $h(n) = n - 1$. Also the identity map $Id : \mathbb{P} \to \mathbb{N}$ is an isomorphism of $(\mathbb{P}; <_P)$ into $(\mathbb{N}; <_N)$. Because of this last fact, we say that $(\mathbb{P}; <_P)$ is a *substructure* of $(\mathbb{N}; <_N)$.

More generally consider two structures $\mathfrak{A}$ and $\mathfrak{B}$ for the language such that $|\mathfrak{A}| \subseteq |\mathfrak{B}|$. It is clear from the definition of homomorphism that the identity map from $|\mathfrak{A}|$ into $|\mathfrak{B}|$ is an isomorphism of $\mathfrak{A}$ into $\mathfrak{B}$ iff

(a) $P^{\mathfrak{A}}$ is the restriction of $P^{\mathfrak{B}}$ to $|\mathfrak{A}|$, for each predicate parameter P;

(b) $f^{\mathfrak{A}}$ is the restriction of $f^{\mathfrak{B}}$ to $|\mathfrak{A}|$, for each function symbol f, and $c^{\mathfrak{A}} = c^{\mathfrak{B}}$ for each constant symbol c.

If these conditions are met, then $\mathfrak{A}$ is said to be a *substructure* of $\mathfrak{B}$, and $\mathfrak{B}$ is an *extension* of $\mathfrak{A}$.

For example, in a language with a two-place function symbol $+$, the structure $(\mathbb{Q}; +_Q)$ is a substructure of $(\mathbb{C}; +_C)$. Here $+_C$ is the addition operation on complex numbers. And $+_Q$, addition on the rationals, is exactly the restriction of $+_C$ to the set $\mathbb{Q}$.

In this example, the set $\mathbb{Q}$ is closed under $+_C$; that is, the sum of two rational numbers is rational. More generally, whenever $\mathfrak{A}$ is a substructure of $\mathfrak{B}$, then $|\mathfrak{A}|$ must be closed under $f^{\mathfrak{B}}$ for every function symbol f. After all, $f^{\mathfrak{B}}(\vec{a})$ (where $\vec{a} \in |\mathfrak{A}|^n$) is nothing but $f^{\mathfrak{A}}(\vec{a})$, which must be some element in $|\mathfrak{A}|$. This closure property even holds for the 0-place function symbols; $c^{\mathfrak{B}}$ must belong to $|\mathfrak{A}|$ for each constant symbol c.

Conversely, suppose we have a structure $\mathfrak{B}$, and let A be any non-empty subset of $|\mathfrak{B}|$ that is closed under all of $\mathfrak{B}$'s functions, as in the preceding paragraph. Then we can make a substructure of $\mathfrak{B}$ with universe A. In fact there is only one way to do this. The universe is A, each predicate parameter P is assigned the restriction of $P^{\mathfrak{B}}$ to A, and similarly for the function symbols. As an extreme case, if the language has no function symbols at all (not even constant symbols), then we can make a substructure out of *any* nonempty subset A of $|\mathfrak{B}|$.

These are basically algebraic concepts, but the following theorem relates them to the logical concepts of truth and satisfaction.

> **HOMOMORPHISM THEOREM**   Let h be a homomorphism of $\mathfrak{A}$ into $\mathfrak{B}$, and let s map the set of variables into $|\mathfrak{A}|$.
>
> (a) For any term t, we have $h(\bar{s}(t)) = \overline{h \circ s}(t)$, where $\bar{s}(t)$ is computed in $\mathfrak{A}$ and $\overline{h \circ s}(t)$ is computed in $\mathfrak{B}$.
>
> (b) For any quantifier-free formula α not containing the equality symbol,
>
> $$\models_{\mathfrak{A}} \alpha[s] \quad \text{iff} \quad \models_{\mathfrak{B}} \alpha[h \circ s].$$
>
> (c) If h is one-to-one (i.e., is an isomorphism of $\mathfrak{A}$ into $\mathfrak{B}$), then in part (b) we may delete the restriction "not containing the equality symbol."
>
> (d) If h is a homomorphism of $\mathfrak{A}$ *onto* $\mathfrak{B}$, then in (b) we may delete the restriction "quantifier-free."

PROOF.   Part (a) uses induction on t; see Exercise 13. Note that $h \circ s$ maps the set of variables into $|\mathfrak{B}|$; its extension to the set of all terms is $\overline{h \circ s}$. It is $\overline{h \circ s}$ that is here being evaluated at t.

(b) For an atomic formula such as Pt, we have

$$\models_{\mathfrak{A}} Pt[s] \Leftrightarrow \bar{s}(t) \in P^{\mathfrak{A}}$$
$$\Leftrightarrow h(\bar{s}(t)) \in P^{\mathfrak{B}} \quad \text{since } h \text{ is a homomorphism}$$
$$\Leftrightarrow \overline{h \circ s}(t) \in P^{\mathfrak{B}} \quad \text{by (a)}$$
$$\Leftrightarrow \models_{\mathfrak{B}} Pt[h \circ s].$$

An inductive argument is then required to handle the connective symbols $\neg$ and $\rightarrow$, but it is completely routine.

(c) In any case,

$$\models_{\mathfrak{A}} u = t[s] \Leftrightarrow \bar{s}(u) = \bar{s}(t)$$
$$\Rightarrow h(\bar{s}(u)) = h(\bar{s}(t))$$
$$\Leftrightarrow \overline{h \circ s}(u) = \overline{h \circ s}(t) \quad \text{by (a)}$$
$$\Leftrightarrow \models_{\mathfrak{B}} u = t[h \circ s].$$

If h is one-to-one, the arrow in the second step can be reversed.

(d) We must extend the routine inductive argument of part (b) to include the quantifier step. That is, we must show that if φ has

the property that for every s,

$$\models_{\mathfrak{A}} \varphi[s] \Leftrightarrow \models_{\mathfrak{B}} \varphi[h \circ s],$$

then $\forall x \, \varphi$ enjoys the same property. We have in any case (as a consequence of the inductive hypothesis on φ) the implication

$$\models_{\mathfrak{B}} \forall x \, \varphi[h \circ s] \Rightarrow \models_{\mathfrak{A}} \forall x \, \varphi[s].$$

This is intuitively very plausible; if φ is true of everything in the larger set $|\mathfrak{B}|$, then *a fortiori* it is true of everything in the smaller set ran h. The details are, for an element a of $|\mathfrak{A}|$,

$$\models_{\mathfrak{B}} \forall x \, \varphi[h \circ s] \Rightarrow \models_{\mathfrak{B}} \varphi[(h \circ s)(x \mid h(a))]$$

$$\Leftrightarrow \models_{\mathfrak{B}} \varphi[h \circ (s(x \mid a))], \qquad \text{the functions being the same}$$

$$\Leftrightarrow \models_{\mathfrak{A}} \varphi[s(x \mid a)] \qquad \text{by the inductive hypothesis.}$$

Now for the converse, suppose that $\not\models_{\mathfrak{B}} \forall x \, \varphi[h \circ s]$, so that $\models_{\mathfrak{B}} \neg \varphi[(h \circ s)(x \mid b)]$ for some element b in $|\mathfrak{B}|$. We need the implication

($*$) *If* for some b in $|\mathfrak{B}|$, $\models_{\mathfrak{B}} \neg \varphi[(h \circ s)(x \mid b)]$, *then* for some a in $|\mathfrak{A}|$, $\models_{\mathfrak{B}} \neg \varphi[(h \circ s)(s \mid h(a))]$.

For given ($*$), we can proceed:

$$\models_{\mathfrak{B}} \neg \varphi[(h \circ s)(x \mid h(a))] \Leftrightarrow \models_{\mathfrak{B}} \neg \varphi[h \circ (s(x \mid a))], \qquad \text{the functions being the same}$$

$$\Leftrightarrow \models_{\mathfrak{A}} \neg \varphi[s(x \mid a)] \qquad \text{by the inductive hypothesis}$$

$$\Rightarrow \not\models_{\mathfrak{A}} \forall x \, \varphi[s].$$

If h maps $|\mathfrak{A}|$ *onto* $|\mathfrak{B}|$, then ($*$) is immediate; we take a such that $b = h(a)$. (But there might be other fortunate times when ($*$) can be asserted even if h fails to have range $|\mathfrak{B}|$.) $\dashv$

Two structures $\mathfrak{A}$ and $\mathfrak{B}$ for the language are said to be *elementarily equivalent* (written $\mathfrak{A} \equiv \mathfrak{B}$) iff for any sentence σ,

$$\models_{\mathfrak{A}} \sigma \Leftrightarrow \models_{\mathfrak{B}} \sigma.$$

COROLLARY 22D   Isomorphic structures are elementarily equivalent:

$$\mathfrak{A} \cong \mathfrak{B} \Rightarrow \mathfrak{A} \equiv \mathfrak{B}$$

Actually more is true. Isomorphic structures are alike in every "structural" way; not only do they satisfy the same first-order sentences, they also satisfy the same second-order (and higher) sentences (i.e., they are secondarily equivalent and more).

There are elementarily equivalent structures that are not isomorphic. For example, it can be shown that the structure $(\mathbb{R}; <_R)$ consisting of the set of real numbers with its usual ordering relation is elementarily

equivalent to the structure $(\mathbb{Q}; <_Q)$ consisting of the set of rational numbers with its ordering (see Section 2.6). But $\mathbb{Q}$ is a countable set whereas $\mathbb{R}$ is not, so these structures cannot be isomorphic. In Section 2.6 we will see how easy it is to make elementarily equivalent structures of differing cardinalities.

> EXAMPLE, revisited.   We had an isomorphism h from $(\mathbb{P}; <_P)$ onto $(\mathbb{N}; <_N)$. So in particular, $(\mathbb{P}; <_P) \equiv (\mathbb{N}; <_N)$; these structures are indistinguishable by first-order sentences.
>
> We furthermore noted that the identity map was an isomorphic embedding of $(\mathbb{P}; <_P)$ into $(\mathbb{N}; <_N)$. Hence for a function $s\colon V \to \mathbb{P}$ and a quantifier-free φ,
>
> $$\models_{(\mathbb{P}; <_P)} \varphi[s] \Leftrightarrow \models_{(\mathbb{N}; <_N)} \varphi[s].$$
>
> This equivalence may fail if φ contains quantifiers. For example,
>
> $$\models_{(\mathbb{P}; <_P)} \forall v_2 (v_1 \neq v_2 \to v_1 < v_2)[\![1]\!],$$
>
> but
>
> $$\nvDash_{(\mathbb{N}; <_N)} \forall v_2 (v_1 \neq v_2 \to v_1 < v_2)[\![1]\!].$$

An *automorphism* of the structure $\mathfrak{A}$ is an isomorphism of $\mathfrak{A}$ onto $\mathfrak{A}$. The identity function on $|\mathfrak{A}|$ is trivially an automorphism of $\mathfrak{A}$. $\mathfrak{A}$ may or may not have nontrivial automorphisms. (We say that $\mathfrak{A}$ is *rigid* if the identity function is its only automorphism.) As a consequence of the homomorphism theorem, we can show that an automorphism must preserve the definable relations:

> COROLLARY 22E   Let h be an automorphism of the structure $\mathfrak{A}$, and let R be an n-ary relation on $|\mathfrak{A}|$ definable in $\mathfrak{A}$. Then for any $a_1, \ldots, a_n$ in $|\mathfrak{A}|$,
>
> $$\langle a_1, \ldots, a_n \rangle \in R \;\Leftrightarrow\; \langle h(a_1), \ldots, h(a_n) \rangle \in R.$$

> PROOF.   Let φ be a formula that defines R in $\mathfrak{A}$. We need to know that
>
> $$\models_{\mathfrak{A}} \varphi[\![a_1, \ldots, a_n]\!] \Leftrightarrow \models_{\mathfrak{A}} \varphi[\![h(a_1), \ldots, h(a_n)]\!].$$
>
> But this is immediate from the homomorphism theorem.   ⊣

This corollary is sometimes useful in showing that a given relation is *not* definable. Consider, for example, the structure $(\mathbb{R}; <)$ consisting of the set of real numbers with its usual ordering. An automorphism of this structure is simply a function h from $\mathbb{R}$ onto $\mathbb{R}$ that is strictly increasing:

$$a < b \Leftrightarrow h(a) < h(b).$$

One such automorphism is the function h for which $h(a) = a^3$. Since this function maps points outside of $\mathbb{N}$ into $\mathbb{N}$, the set $\mathbb{N}$ is not definable in this structure.

Another example is provided by elementary algebra books, which sometimes explain that the length of a vector in the plane cannot be defined in terms of vector addition and scalar multiplication. For the map that takes a vector $\mathbf{x}$ into the vector $2\mathbf{x}$ is an automorphism of the plane with respect to vector addition and scalar multiplication, but it is not length-preserving. From our viewpoint, the structure in question,

$$(E; +, f_r)_{r \in \mathbb{R}},$$

has for its universe the plane E, has the binary operation $+$ of vector addition, and has (for each r in the set $\mathbb{R}$) the unary operation f_r of scalar multiplication by r. (Thus the language in question has a one-place function symbol for each real number.) The doubling map described above is an automorphism of this structure. But it does not preserve the set of unit vectors,

$$\{\mathbf{x} \mid \mathbf{x} \in E \ \text{and } \mathbf{x} \text{ has length } 1\}.$$

So this set cannot be definable in the structure. (Incidentally, the homomorphisms of vector spaces are called *linear transformations*.)

Exercises

1. Show that (a) $\Gamma; \alpha \models \varphi$ iff $\Gamma \models (\alpha \rightarrow \varphi)$; and (b) $\varphi \models\!\models \psi$ iff $\models (\varphi \leftrightarrow \psi)$.

2. Show that no one of the following sentences is logically implied by the other two. (This is done by giving a structure in which the sentence in question is false, while the other two are true.)
   (a) $\forall x \,\forall y \,\forall z (Pxy \rightarrow Pyz \rightarrow Pxz)$. Recall that by our convention $\alpha \rightarrow \beta \rightarrow \gamma$ is $\alpha \rightarrow (\beta \rightarrow \gamma)$.
   (b) $\forall x \,\forall y (Pxy \rightarrow Pyx \rightarrow x = y)$.
   (c) $\forall x \,\exists y \, Pxy \rightarrow \exists y \,\forall x \, Pxy$.

3. Show that
$$\{\forall x(\alpha \rightarrow \beta), \forall x\, \alpha\} \models \forall x\, \beta.$$

4. Show that if x does not occur free in α, then $\alpha \models \forall x\, \alpha$.

5. Show that the formula $x = y \rightarrow Pzfx \rightarrow Pzfy$ (where f is a one-place function symbol and P is a two-place predicate symbol) is valid.

6. Show that a formula θ is valid iff $\forall x\, \theta$ is valid.

7. Restate the definition of "$\mathfrak{A}$ satisfies φ with s" in the way described

on page 84. That is, define by recursion a function $\bar{h}$ such that $\mathfrak{A}$ satisfies φ with s iff $s \in \bar{h}(\varphi)$.

8. Assume that Σ is a set of sentences such that for any sentence τ, either $\Sigma \models \tau$ or $\Sigma \models \neg \tau$. Assume that $\mathfrak{A}$ is a model of Σ. Show that for any sentence τ, we have $\models_{\mathfrak{A}} \tau$ iff $\Sigma \models \tau$.

9. Assume that the language has equality and a two-place predicate symbol P. For each of the following conditions, find a sentence σ such that the structure $\mathfrak{A}$ is a model of σ iff the condition is met.
   (a) $|\mathfrak{A}|$ has exactly two members.
   (b) $P^{\mathfrak{A}}$ is a function from $|\mathfrak{A}|$ into $|\mathfrak{A}|$. (A *function* is a single-valued relation, as in Chapter 0. For f to be a function from A into B, the domain of f must be all of A; the range of f is a subset, not necessarily proper, of B.)
   (c) $P^{\mathfrak{A}}$ is a permutation of $|\mathfrak{A}|$; i.e., $P^{\mathfrak{A}}$ is a one-to-one function with domain and range equal to $|\mathfrak{A}|$.

10. Show that

$$\models_{\mathfrak{A}} \forall v_2 \, Q v_1 v_2 [\![c^{\mathfrak{A}}]\!] \quad \text{iff} \quad \models_{\mathfrak{A}} \forall v_2 \, Q c v_2.$$

   Here Q is a two-place predicate symbol and c is a constant symbol.

11. For each of the following relations, give a formula which defines it in $(\mathbb{N}; +, \cdot)$. (The language is assumed to have equality and the parameters $\forall$, $+$, and $\cdot$).
   (a) $\{0\}$.
   (b) $\{1\}$.
   (c) $\{\langle m, n \rangle \mid n$ is the successor of m in $N\}$.
   (d) $\{\langle m, n \rangle \mid m < n$ in $\mathbb{N}\}$.
      Digression: This is merely the tip of the iceberg. A relation on $\mathbb{N}$ is said to be *arithmetical* if it is definable in this structure. All decidable relations are arithmetical, as are many others. The arithmetical relations can be arranged in a hierarchy; see Section 3.5.

12. Let $\mathfrak{R}$ be the structure $(\mathbb{R}; +, \cdot)$. (The language is assumed to have equality and the parameters $\forall$, $+$, and $\cdot$. $\mathfrak{R}$ is the structure whose universe is the set $\mathbb{R}$ of real numbers and such that $+^{\mathfrak{R}}$ and $\cdot^{\mathfrak{R}}$ are the usual addition and multiplication operations.)
   (a) Give a formula that defines in $\mathfrak{R}$ the interval $[0, \infty)$.
   (b) Give a formula that defines in $\mathfrak{R}$ the set $\{2\}$.
   *(c) Show that any finite union of intervals, the endpoints of which are algebraic, is definable in $\mathfrak{R}$. (The converse is also true; these are the only definable sets in the structure. But we will not prove this fact.)

13. Prove part (a) of the homomorphism theorem.

14. What subsets of the real line $\mathbb{R}$ are definable in $(\mathbb{R}; <)$? What subsets of the plane $\mathbb{R} \times \mathbb{R}$ are definable in $(\mathbb{R}; <)$?

    Remarks: The nice thing about $(\mathbb{R}; <)$ is that its automorphisms are exactly the order-preserving maps from $\mathbb{R}$ onto itself. But stop after the binary relations. There are 2^{13} definable ternary relations, so you do not want to catalog all of them.

15. Show that the addition relation, $\{\langle m, n, p \rangle \mid p = m + n\}$, is not definable in $(\mathbb{N}; \cdot)$. *Suggestion*: Consider an automorphism of $(\mathbb{N}; \cdot)$ that switches two primes.

    Digression: Algebraically, the structure of the natural numbers with multiplication is nothing but the free Abelian semigroup with $\aleph_0$ generators (viz. the primes), together with a zero element. There is no way you could define addition here. If you could define addition, then you could define ordering (by Exercise 11 and the natural transitivity statement). But one generator looks just like another. That is, there are $2^{\aleph_0}$ automorphisms — simply permute the primes. None of them is order-preserving except the identity.

16. Give a sentence having models of size $2n$ for every positive integer n, but no finite models of odd size. (Here the language should include equality and will have whatever parameters you choose.) *Suggestion*: One method is to make a sentence that says, "Everything is either red or blue, and f is a color-reversing permutation."

    Remark: Given a sentence σ, it might have some finite models (i.e., models with finite universes). Define the *spectrum* of σ to be the set of positive integers n such that σ has a model of size n. This exercise shows that the set of even numbers is a spectrum.

    For example if σ is the conjunction of the field axioms (there are only finitely many, so we can take their conjunction), then its spectrum is the set of powers of primes. This fact is proved in any course on finite fields. The spectrum of $\neg \sigma$, by contrast, is the set of all positive integers (non-fields come in all sizes).

    Günter Asser in 1955 raised the question: Is the complement of every spectrum a spectrum? Once you realize that simply taking a negation does not work (cf. the preceding paragraph), you see that this is a nontrivial question. In fact the problem, known as the spectrum problem, is still open. But modern work has tied it to another open problem, whether or not co-NP $=$ NP.

17. (a) Consider a language with equality whose only parameter (aside from $\forall$) is a two-place predicate symbol P. Show that if $\mathfrak{A}$ is finite and $\mathfrak{A} \equiv \mathfrak{B}$, then $\mathfrak{A}$ is isomorphic to $\mathfrak{B}$. *Suggestion*: Suppose the universe of $\mathfrak{A}$ has size n. Make a single sentence σ of the form $\exists v_1 \cdots \exists v_n \, \theta$ that describes $\mathfrak{A}$ "completely." That is, on the one hand, σ must be true in $\mathfrak{A}$. And on the other hand, any model of σ must be exactly like (i.e., isomorphic to) $\mathfrak{A}$.

   *(b) Show that the result of part (a) holds regardless of what param-
         eters the language contains.

18. A universal ($\forall_1$) formula is one of the form $\forall x_1 \cdots \forall x_n \, \theta$, where
    θ is quantifier-free. An existential ($\exists_1$) formula is of the dual form
    $\exists x_1 \cdots \exists x_n \, \theta$. Let $\mathfrak{A}$ be a substructure of $\mathfrak{B}$, and let $s : V \to |\mathfrak{A}|$.
    (a) Show that if $\models_{\mathfrak{A}} \psi[s]$ and ψ is existential, then $\models_{\mathfrak{B}} \psi[s]$. And
         if $\models_{\mathfrak{B}} \varphi[s]$ and φ is universal, then $\models_{\mathfrak{A}} \varphi[s]$.
    (b) Conclude that the sentence $\exists x \, Px$ is not logically equivalent to
         any universal sentence, nor $\forall x \, Px$ to any existential sentence.
               Remark: Part (a) says (when φ is a sentence) that any univer-
         sal sentence is "preserved under substructures." Being universal
         is a syntactic property — it has to do with the string of symbols.
         In contrast, being preserved under substructures is a semantic
         property — it has to do with satisfaction in structures. But this
         semantic property captures the syntactic property up to logical
         equivalence (which is all one could ask for). That is, if σ is a
         sentence that is always preserved under substructures, then σ
         is logically equivalent to a universal sentence. (This fact is due
         to Łoś and Tarski.)

19. An $\exists_2$ formula is one of the form $\exists x_1 \cdots \exists x_n \, \theta$, where θ is
    universal.
    (a) Show that if an $\exists_2$ sentence in a language not containing func-
         tion symbols (not even constant symbols) is true in $\mathfrak{A}$, then it
         is true in some finite substructure of $\mathfrak{A}$.
    (b) Conclude that $\forall x \, \exists y \, Pxy$ is not logically equivalent to any $\exists_2$
         sentence.

20. Assume the language has equality and a two-place predicate symbol
    P. Consider the two structures $(\mathbb{N}; <)$ and $(\mathbb{R}; <)$ for the language.
    (a) Find a sentence true in one structure and false in the other.
    *(b) Show that any $\exists_2$ sentence (as defined in the preceding exer-
         cise) true in $(\mathbb{R}; <)$ is also true in $(\mathbb{N}; <)$. *Suggestion*: First,
         for any finite set of real numbers, there is an automorphism
         of $(\mathbb{R}; <)$ taking those real numbers to natural numbers. Sec-
         ondly, by Exercise 18, universal formulas are preserved under
         substructures.

21. We could consider enriching the language by the addition of a new
    quantifier. The formula $\exists! x \, \alpha$ (read "there exists a unique x such
    that α") is to be satisfied in $\mathfrak{A}$ by s iff there is one and only one
    $a \in |\mathfrak{A}|$ such that $\models_{\mathfrak{A}} \alpha[s(x \mid a)]$. Assume that the language has the
    equality symbol and show that this apparent enrichment comes to
    naught, in the sense that we can find an ordinary formula logically
    equivalent to $\exists! x \, \alpha$.

22. Assume that $\mathfrak{A}$ is a structure and h is a function with $\operatorname{ran} h = |\mathfrak{A}|$.

Show that there is a structure $\mathfrak{B}$ such that h is a homomorphism of $\mathfrak{B}$ onto $\mathfrak{A}$. *Suggestion*: We need to take $|\mathfrak{B}| = \operatorname{dom} h$. In general, the axiom of choice will be needed to define the functions in $\mathfrak{B}$, unless h is one-to-one.

Remark: The result yields an "upward Löwenheim–Skolem theorem without equality" (cf. Section 2.6). That is, any structure $\mathfrak{A}$ has an extension to a structure $\mathfrak{B}$ of any higher cardinality such that $\mathfrak{A}$ and $\mathfrak{B}$ are elementarily equivalent, except for equality. There is nothing deep about this. Not until you add equality.

23. Let $\mathfrak{A}$ be a structure and g a one-to-one function with $\operatorname{dom} g = |\mathfrak{A}|$. Show that there is a unique structure $\mathfrak{B}$ such that g is an isomorphism of $\mathfrak{A}$ onto $\mathfrak{B}$.

24. Let h be an isomorphic embedding of $\mathfrak{A}$ into $\mathfrak{B}$. Show that there is a structure $\mathfrak{C}$ isomorphic to $\mathfrak{B}$ such that $\mathfrak{A}$ is a substructure of $\mathfrak{C}$. *Suggestion*: Let g be a one-to-one function with domain $|\mathfrak{B}|$ such that $g(h(a)) = a$ for $a \in |\mathfrak{A}|$. Form $\mathfrak{C}$ such that g is an isomorphism of $\mathfrak{B}$ onto $\mathfrak{C}$.

Remark: The result stated in this exercise should not seem surprising. On the contrary, it is one of those statements that is obvious until you have to prove it. It says that if you can embed $\mathfrak{A}$ isomorphically into $\mathfrak{B}$, then for all practical purposes you can pretend $\mathfrak{A}$ is a substructure of $\mathfrak{B}$.

25. Consider a fixed structure $\mathfrak{A}$. Expand the language by adding a new constant symbol c_a for each $a \in |\mathfrak{A}|$. Let $\mathfrak{A}^+$ be the structure for this expanded language that agrees with $\mathfrak{A}$ on the original parameters and that assigns to c_a the point a. A relation R on $|\mathfrak{A}|$ is said to be *definable from points* in $\mathfrak{A}$ iff R is definable in $\mathfrak{A}^+$. (This differs from ordinary definability only in that we now have parameters in the language for members of $|\mathfrak{A}|$.) Let $\mathfrak{R} = (\mathbb{R}; <, +, \cdot)$.
    (a) Show that if A is a subset of $\mathbb{R}$ consisting of the union of finitely many intervals, then A is definable from points in $\mathfrak{R}$ (cf. Exercise 12).
    (b) Assume that $\mathfrak{A} \equiv \mathfrak{R}$. Show that any subset of $|\mathfrak{A}|$ that is nonempty, bounded (in the ordering $<^{\mathfrak{A}}$), and definable from points in $\mathfrak{A}$ has a least upper bound in $|\mathfrak{A}|$.

Digression: Often when people speak of definability within a structure, this is the concept they mean. The more standard phrase is "definable from parameters"; here "points" is used because the word "parameter" is used in a different sense in this chapter.

The real ordered field can be characterized up to isomorphism by saying that it is a complete ordered field. (This fact should be included in any analysis course.) But completeness (i.e., that nonempty bounded sets have least upper bounds) is not a first-

order property. See Example 4 in Section 4.1 for its second-order statement. The first-order "image" of completeness is given by the schema obtained from that second-order statement by replacing X by a first-order formula φ. The resulting schema (i.e., the set of sentences you get by letting φ vary and taking universal closure) says that the least-upper-bound property holds for the sets that are definable from points. Ordered fields satisfying those sentences are called "real closed-ordered fields."

The surprising fact is that such fields were not invented by logicians. They were previously studied by algebraists and you can read about them in van der Waerden's *Modern Algebra* book (volume I). Of course, he uses a characterization of them that does not involve logic.

What Tarski showed is that any real closed-ordered field is elementarily equivalent to the field of real numbers. From this it follows that the theory of the real-ordered field is decidable.

26. (a) Consider a fixed structure $\mathfrak{A}$ and define its *elementary type* to be the class of structures elementarily equivalent to $\mathfrak{A}$. Show that this class is EC_Δ. *Suggestion*: Show it is Mod Th $\mathfrak{A}$.
    (b) Call a class $\mathcal{K}$ of structures *elementarily closed* or ECL if whenever a structure belongs to $\mathcal{K}$ then all elementarily equivalent structures also belong. Show that any such class is a union of EC_Δ classes. (A class that is a union of EC_Δ classes is said to be an $\mathrm{EC}_{\Delta\Sigma}$ class; this notation is derived from topology.)
    (c) Conversely, show that any class that is the union of EC_Δ classes is elementarily closed.

27. Assume that the parameters of the language are $\forall$ and a two-place predicate symbol P. List all of the non-isomorphic structures of size 2. That is, give a list of structures (where the universe of each has size 2) such that any structure of size 2 is isomorphic to exactly one structure on the list.

28. For each of the following pairs of structures, show that they are not elementarily equivalent, by giving a sentence true in one and false in the other. (The language here contains $\forall$ and a two-place function symbol $\circ$.)
    (a) $(\mathbb{R}; \times)$ and $(\mathbb{R}^*; \times^*)$, where $\times$ is the usual multiplication operation on the real numbers, $\mathbb{R}^*$ is the set of non-zero reals, and $\times^*$ is $\times$ restricted to the non-zero reals.
    (b) $(\mathbb{N}; +)$ and $(\mathbb{P}; +^*)$, where $\mathbb{P}$ is the set of positive integers, and $+^*$ is usual addition operation restricted to $\mathbb{P}$.
    (c) Better yet, for each of the four structures of parts (a) and (b), give a sentence true in that structure and false in the other three.

SECTION 2.3

A Parsing Algorithm[1]

As in sentential logic, we need to know that we can decompose formulas (and terms) in a unique way, discovering how they are built up. The uniqueness is necessary to justify our definitions by recursion, such as the definition of satisfaction in the preceding section.

For terms we used Polish notation; for formulas we relied on parentheses. Accordingly, we first consider a decomposition procedure for terms, showing unique readability. And then we extend the methods to cover the formulas.

Recall that the terms are built up from the variables and constant symbols by operations corresponding to the function symbols. We now define a function K on the symbols involved such that for a symbol s, $K(s) = 1 - n$, where n is the number of terms that must follow s to obtain a term:

$$K(x) = 1 - 0 = 1 \quad \text{for a variable } x;$$
$$K(c) = 1 - 0 = 1 \quad \text{for a constant symbol } c;$$
$$K(f) = 1 - n \quad\quad\ \text{for an } n\text{-place function symbol } f.$$

We then extend K to the set of expressions using these symbols by defining

$$K(s_1 s_2 \cdots s_n) = K(s_1) + K(s_2) + \cdots + K(s_n).$$

Since no symbol is a finite sequence of others, this definition is unambiguous.

LEMMA 23A   For any term t, $K(t) = 1$.

PROOF.   Use induction on t. The inductive step, for an n-place function symbol f, is

$$K(f t_1 \cdots t_n) = (1 - n) + \underbrace{(1 + \cdots + 1)}_{n \text{ times}} = 1.$$

$\dashv$

In fact, K was chosen to be the unique function on these symbols for which Lemma 23A holds. It follows from this lemma that if ε is a concatenation of m terms, then $K(\varepsilon) = m$.

By a *terminal* segment of a string $\langle s_1, \ldots, s_n \rangle$ of symbols we mean a sequence of the form $\langle s_k, s_{k+1}, \ldots, s_n \rangle$, where $1 \le k \le n$.

LEMMA 23B   Any terminal segment of a term is a concatenation of one or more terms.

---

[1] This section may be omitted by a reader willing to accept the meaningfulness of our many definitions by recursion.

PROOF.   We use induction on the term. For a one-symbol term (i.e., a variable or a constant symbol) the conclusion follows trivially. For a term $f t_1 \cdots t_n$, any terminal segment (other than the term itself) must equal

$$t'_k t_{k+1} \cdots t_n,$$

where $k \leq n$ and t'_k is a terminal segment of t_k. By the inductive hypothesis t'_k is a concatenation of, say, m terms, where $m \geq 1$. So altogether we have $m + (n - k)$ terms.   ⊣

COROLLARY 23C   No proper initial segment of a term is itself a term. If t_1 is a proper initial segment of a term t, then $K(t_1) < 1$.

PROOF.   Suppose a term t is divided into a proper initial segment t_1 and a terminal segment t_2. Then $1 = K(t) = K(t_1) + K(t_2)$, and by Lemma 23B, $K(t_2) \geq 1$. Hence $K(t_1) < 1$ and t_1 cannot be a term.   ⊣

Parsing Terms

We want an algorithm that, given an expression, will determine whether or not the expression is a legal term, and if it is, will construct the unique tree showing how the term is built up.

Assume then that we are given an expression. We will construct a tree, with the given expression at the top (i.e., the root). Initially it is the only vertex in the tree, but as the procedure progresses, the tree will grow downward.

The algorithm consist of the following two steps.

1. If each minimal vertex (at the bottom) has a single symbol (which must* be a variable or a constant symbol), then the procedure is completed. (The given expression is indeed a term, and we have constructed its tree.) Otherwise, select a minimal vertex having an expression with two or more symbols. We examine that expression.

2. The first symbol must* be an n-place function symbol, say f, where $n > 0$. We extend the tree downward by creating n new vertices below the present one. Scan the expression after the f, until first reaching a string t (of variables, constant symbols, and function symbols) with $K(t) = 1$.** Then t is the expression that goes at the leftmost unlabeled new vertex. Repeat with the remainder of the expression, until all n new vertices have been labeled, and the expression has been exhausted.** Return to step 1.

---

* If not, then the expression here is not a term. We reject the given expression as a non-term, and halt.

** If the end of the expression is reached before finding such an t, then the expression here is not a term. We reject the given expression as a non-term, and halt.

As in Section 1.3, the crucial point is that the tree could not have been made differently. In step 2, we selected the first string t we found with $K(t) = 1$. We could not have used less than t (because we needed $K(t) = 1$ by Lemma 23A). We could not have used more than t (because that longer string would have the proper initial segment t with $K(t) = 1$, contradicting Corollary 23C). The choice of t was the only possible choice.

When the algorithm halts, either it has rejected the given expression as a non-term, or it has constructed the one tree demonstrating that the given expression is a legal tree.

We can rephrase the uniqueness in the terminology of Section 1.4 as follows:

> **UNIQUE READABILITY THEOREM FOR TERMS**   The set of terms is freely generated from the set of variables and constant symbols by the $\mathcal{F}_f$ operations.

> PROOF.   First, it is clear that iff $f \neq g$, then ran $\mathcal{F}_f$ is disjoint from ran $\mathcal{F}_g$; this requires checking only the first symbol. Furthermore, both ranges are disjoint from the set of variables and constant symbols. It remains only to show that $\mathcal{F}_f$, when restricted to terms, is one-to-one. Suppose, for a two-place f, we have
>
> $$f t_1 t_2 = f t_3 t_4.$$
>
> By deleting the first symbol we are left with
>
> $$t_1 t_2 = t_3 t_4.$$
>
> If $t_1 \neq t_3$, then one would be a proper initial segment of the other, which is impossible for terms by Corollary 23C. So $t_1 = t_3$, and we are then left with $t_2 = t_4$.   ⊣

Parsing Formulas

To extend this argument to formulas, we now define K on the other symbols:

$$K(() = -1;$$
$$K()) = 1;$$
$$K(\forall) = -1;$$
$$K(\neg) = 0;$$
$$K(\rightarrow) = -1;$$
$$K(P) = 1 - n \quad \text{for an } n\text{-place predicate symbol } P;$$
$$K(=) = -1.$$

The idea behind the definition is again that $K(s)$ should be $1 - n$, where n is the number of things (right parentheses, terms, or formulas) required to go after s. We extend K as usual to the set of all expressions:

$$K(s_1 \cdots s_n) = K(s_1) + \cdots + K(s_n).$$

LEMMA 23D   For any wff α, $K(\alpha) = 1$.

PROOF.   Another straightforward induction.   ⊣

LEMMA 23E   For any proper initial segment α' of a wff α, $K(\alpha') < 1$.

PROOF.   Use induction on α. The details are left to Exercise 1.   ⊣

COROLLARY 23F   No proper initial segment of a formula is itself a formula.

Armed with these facts, we can proceed as in Section 1.3. Instead of sentence symbols at the minimal vertices, we now have atomic formulas (which are distinguished by having first an n-place predicate symbol, followed by n terms).

A wff that is non-atomic must begin either with $\forall v_i$ or with (. In the former case we have one new vertex; in the latter case we need to examine the next symbol to see if it is $\neg$. If not, then we can either count parentheses or use the K function — both methods work — to find the correct split.

Again, the uniqueness can be phrased in the terminology of Section 1.4 as follows:

UNIQUE READABILITY THEOREM FOR FORMULAS   The set of wffs is freely generated from the set of atomic formulas by the operations $\mathcal{E}_\neg$, $\mathcal{E}_\rightarrow$, and $\mathcal{Q}_i$ ($i = 1, 2, \ldots$).

PROOF.   The unary operations $\mathcal{E}_\neg$ and $\mathcal{Q}_i$ are obviously one-to-one. As in Section 1.4, we can show that the restriction of $\mathcal{E}_\rightarrow$ to wffs is one-to-one.

The disjointness half of the theorem follows from the ad hoc observations:

1. ran $\mathcal{E}_\neg$, ran $\mathcal{Q}_i$, ran $\mathcal{Q}_j$, and the set of atomic formulas are pairwise disjoint, for $i \neq j$. (Just look at the first two symbols.)
2. ran $\mathcal{E}_\rightarrow$, ran $\mathcal{Q}_i$, ran $\mathcal{Q}_j$, and the set of atomic formulas are similarly pairwise disjoint, for $i \neq j$.
3. For a wff β, $(\neg \alpha) \neq (\beta \rightarrow \gamma)$, because no wff begins with $\neg$. Hence ran $\mathcal{E}_\neg$ is disjoint from the range of the restriction of $\mathcal{E}_\rightarrow$ to wffs.   ⊣

Exercises

1. Show that for a proper initial segment α' of a wff α, we have $K(\alpha') < 1$.

2. Let ε be an expression consisting of variables, constant symbols, and function symbols. Show that ε is a term iff $K(\varepsilon) = 1$ and for every terminal segment ε' of ε we have $K(\varepsilon') > 0$. *Suggestion*: Prove the

stronger result that if $K(\varepsilon') > 0$ for every terminal segment ε' of ε, then ε is a concatenation of $K(\varepsilon)$ terms. (This algorithm is due to Jáskowski.)

SECTION 2.4

A Deductive Calculus

Suppose that $\Sigma \models \tau$. What methods of proof might be required to demonstrate that fact? Is there necessarily a proof at all?

Such questions lead immediately to considerations of what constitutes a proof. A proof is an argument that you give to someone else and that completely convinces that person of the correctness of your assertion (in this case, that $\Sigma \models \tau$).

Thus a proof should be finitely long, as you cannot give all of an infinite object to another person. If the set Σ of hypotheses is infinite, they cannot all be used. But the compactness theorem for first-order logic (which we will prove in Section 2.5 using the deductive calculus of this section) will ensure the existence of a finite $\Sigma_0 \subseteq \Sigma$ such that $\Sigma_0 \models \tau$.

Another essential feature of a proof (besides its being finite in length) is that it must be possible for someone else (if that person is to be convinced by it) to check the proof to ascertain that it contains no fallacies. This check must be effective; it must be the sort of thing that can be carried out without brilliant flashes of insight on the part of the checker. In particular, the set of proofs from the empty set of hypotheses (i.e., proofs that $\models \tau$) should be decidable. This implies that the set of formulas provable without hypotheses must be effectively enumerable. For one could in principle enumerate provable sentences by generating all strings of symbols and sorting out the proofs from the nonproofs. When a proof is discovered, its last line is entered on the output list. (This issue will be examined more carefully at the end of Section 2.5.) But here again there is a theorem (the enumerability theorem, proved in Section 2.5) stating that, under reasonable conditions, the validities — the set of valid formulas — is indeed effectively enumerable.

Thus the compactness theorem and the enumerability theorem are necessary conditions for satisfactory proofs of logical implication always to exist. Conversely, we claim that these two theorems are sufficient for proofs (of some sort) to exist. For suppose that $\Sigma \models \tau$. By the compactness theorem, then, there is a finite set $\{\sigma_0, \ldots, \sigma_n\} \subseteq \Sigma$ that logically implies τ. Then $\sigma_0 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \tau$ is valid (Exercise 1, Section 2.2). So to demonstrate conclusively that $\Sigma \models \tau$ one need only carry out a finite number of steps in the enumeration of the validities

until $\sigma_0 \to \cdots \to \sigma_n \to \tau$ appears, and then verify that each $\sigma_i \in \Sigma$. (This should be compared with the complex procedure suggested by the original definition of logical implication, discussed in Section 2.2.) The record of the enumeration procedure that produced $\sigma_0 \to \cdots \to \sigma_n \to \tau$ can then be regarded as a *proof* that $\Sigma \models \tau$. As a proof, it should be acceptable to anyone who accepts the correctness of your procedure for enumerating validities.

Against the foregoing general (and slightly vague) discussion, the outline of this section can be described as follows: We will introduce formal proofs but we will call them *deductions*, to avoid confusion with our English-language proofs. These will mirror (in our model of deductive thought) the proofs made by the working mathematician to convince his or her colleagues of certain truths. Then in Section 2.5 we will show that whenever $\Sigma \models \tau$, there is a deduction of τ from Σ (and only then). This will, as is suggested by the foregoing discussion, yield proofs of the compactness theorem and the enumerability theorem. And in the process we will get to see what methods of deduction are adequate to demonstrate that a given sentence, is, in fact, logically implied by certain others. In other words, our goal is to produce a mathematically precise concept of deduction that, in the context of first-order logic, is adequate and correct.

Formal Deductions

We will shortly select an infinite set Λ of formulas to be called logical axioms. And we will have a rule of inference, which will enable us to obtain a new formula from certain others. Then for a set Γ of formulas, the *theorems* of Γ will be the formulas which can be obtained from $\Gamma \cup \Lambda$ by use of the rule of inference (some finite number of times). If φ is a theorem of Γ (written $\Gamma \vdash \varphi$), then a sequence of formulas that records (as explained below) how φ was obtained from $\Gamma \cup \Lambda$ with the rule of inference will be called a deduction of φ from Γ.

The choice of Λ and the choice of the rule (or rules) of inference are far from unique. In this section we are presenting one deductive calculus for first-order logic, chosen from the array of possible calculi. (For example, one can have $\Lambda = \varnothing$ by using many rules of inference. We will take the opposite extreme; our set Λ will be infinite but we will have only one rule of inference.)

Our one rule of inference is traditionally known as *modus ponens*. It is usually stated: From the formulas α and $\alpha \to \beta$ we may infer β:

$$\frac{\alpha, \ \alpha \to \beta}{\beta}.$$

Thus the theorems of a set Γ are the formulas obtainable from $\Gamma \cup \Lambda$ by use of modus ponens some finite number of times.

DEFINITION.   A *deduction of φ from* Γ is a finite sequence $\langle \alpha_0, \ldots, \alpha_n \rangle$ of formulas such that α_n is φ and for each $k \leq n$, either

(a) α_k is in $\Gamma \cup \Lambda$, or
(b) α_k is obtained by modus ponens from two earlier formulas in the sequence; that is, for some i and j less than k, α_j is $\alpha_i \rightarrow \alpha_k$.

If such a deduction exists, we say that φ is *deducible* from Γ, or that φ is a *theorem* of Γ, and we write $\Gamma \vdash \varphi$.

There is another viewpoint that is useful here: A deduction of φ from Γ can be viewed as a *construction sequence*, showing how φ can be obtained from the set $\Gamma \cup \Lambda$ by applying modus ponens zero or more times. (We hesitate to say that φ is built "up" from $\Gamma \cup \Lambda$. Unlike the formula-building operations that produce longer formulas from shorter ones, modus ponens can produce shorter formulas from longer ones.) That is, the set of theorems of Γ is exactly the set of formulas that are obtainable from the "base" set $\Gamma \cup \Lambda$ by applying modus ponens.

The situation here differs from the one discussed in Section 1.4 in two ways, one unimportant and one important. The unimportant difference is that we get the set of theorems by closing under the "partially defined" operation of modus ponens, whose domain consists only of pairs of formulas of the form $\langle \alpha, \alpha \rightarrow \beta \rangle$ (in contrast to the "totally defined" formula-building operations). The more important difference is that the set of theorems is *not* freely generated from $\Gamma \cup \Lambda$ by modus ponens. This reflects the fact that a theorem never has a *unique* deduction. In Sections 1.3 and 2.3, we were concerned to show that for any wff φ, there was a unique tree (which we could effectively calculate) showing how φ was built up by use of formula-building operations. Now the tree is by no means unique, and calculating one such tree is a very different matter from before.

Nevertheless, this viewpoint does yield the following induction principle. We say that a set S of formulas is *closed* under modus ponens if whenever both $\alpha \in S$ and $(\alpha \rightarrow \beta) \in S$ then also $\beta \in S$.

INDUCTION PRINCIPLE   Suppose that S is a set of wffs that includes $\Gamma \cup \Lambda$ and is closed under modus ponens. Then S contains every theorem of Γ.

For example, if the formulas β, γ, and $\gamma \rightarrow \beta \rightarrow \alpha$ are all in $\Gamma \cup \Lambda$, then $\Gamma \vdash \alpha$, as is evidenced by the tree which displays how α was
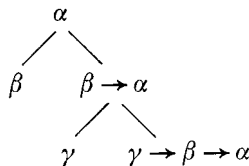
obtained. Although it is tempting (and in some ways more elegant) to define a deduction to be such a tree, it will be simpler to take deductions to be the linear sequences obtained by squashing such trees into straight lines.

Now at last we give the set Λ of logical axioms. These are arranged in six groups. Say that a wff φ is a *generalization* of ψ iff for some $n \geq 0$ and some variables $x_1, \ldots, x_n$,

$$\varphi = \forall x_1 \cdots \forall x_n \psi.$$

We include the case $n = 0$; any wff is a generalization of itself. The logical axioms are then all generalizations of wffs of the following forms, where x and y are variables and α and β are wffs:

1. Tautologies;
2. $\forall x\, \alpha \rightarrow \alpha_t^x$, where t is substitutable for x in α;
3. $\forall x(\alpha \rightarrow \beta) \rightarrow (\forall x\, \alpha \rightarrow \forall x\, \beta)$;
4. $\alpha \rightarrow \forall x\, \alpha$, where x does not occur free in α.

And if the language includes equality, then we add

5. $x = x$;
6. $x = y \rightarrow (\alpha \rightarrow \alpha')$, where α is atomic and α' is obtained from α by replacing x in zero or more (but not necessarily all) places by y.

For the most part groups 3–6 are self-explanatory; we will see various examples later. Groups 1 and 2 require explanation. But first we should admit that the above list of logical axioms may not appear very natural. Later it will be possible to see where each of the six groups originated.

Substitution

In axiom group 2 we find

$$\forall x\, \alpha \rightarrow \alpha_t^x.$$

Here α_t^x is the expression obtained from the formula α by replacing the variable x, wherever it occurs free in α, by the term t. This concept can also be (and for us officially is) defined by recursion:

1. For atomic α, α_t^x is the expression obtained from α by replacing the variable x by t. (This is elaborated upon in Exercise 1. Note that α_t^x is itself a formula.)
2. $(\neg \alpha)_t^x = (\neg \alpha_t^x)$.
3. $(\alpha \rightarrow \beta)_t^x = (\alpha_t^x \rightarrow \beta_t^x)$.
4. $(\forall y\, \alpha)_t^x = \begin{cases} \forall y\, \alpha & \text{if } x = y, \\ \forall y(\alpha_t^x) & \text{if } x \neq y. \end{cases}$

EXAMPLES

1. $\varphi_x^x = \varphi$.
2. $(Qx \rightarrow \forall x\, Px)_y^x = (Qy \rightarrow \forall x\, Px)$.
3. If α is $\neg\forall y\, x = y$, then $\forall x\, \alpha \rightarrow \alpha_z^x$ is

$$\forall x\, \neg\forall y\, x = y \rightarrow \neg\forall y\, z = y.$$

4. For α as in 3, $\forall x\, \alpha \rightarrow \alpha_y^x$ is

$$\forall x\, \neg\forall y\, x = y \rightarrow \neg\forall y\, y = y.$$

The last example above illustrates a hazard which must be guarded against. On the whole, $\forall x\, \alpha \rightarrow \alpha_t^x$ seems like a plausible enough axiom. ("If α is true of everything, then it should be true of t.") But in Example 4, we have a sentence of the form $\forall x\, \alpha \rightarrow \alpha_t^x$, which is nearly always false. The antecedent, $\forall x\, \neg\forall y\, x = y$, is true in any structure whose universe contains two or more elements. But the consequent, $\neg\forall y\, y = y$, is false in any structure. So something has gone wrong.

The problem is that when y was substituted for x, it was immediately "captured" by the $\forall y$ quantifier. We must impose a restriction on axiom group 2 that will preclude this sort of quantifier capture. Informally, we can say that a term t is *not* substitutable for x in α if there is some variable y in t that is captured by a $\forall y$ quantifier in α_t^x. The real definition is given below by recursion. (Since the concept will be used later in proofs by induction, a recursive definition is actually the most usable variety.)

Let x be a variable, t a term. We define the phrase "t is *substitutable* for x in α" as follows:

1. For atomic α, t is always substitutable for x in α. (There are no quantifiers in α, so no capture could occur.)
2. t is substitutable for x in $(\neg\alpha)$ iff it is substitutable for x in α. t is substitutable for x in $(\alpha \rightarrow \beta)$ iff it is substitutable for x in both α and β.
3. t is substitutable for x in $\forall y\, \alpha$ iff either
   (a) x does not occur free in $\forall y\, \alpha$, or
   (b) y does not occur in t and t is substitutable for x in α.
(The point here is to be sure that nothing in t will be captured by the $\forall y$ prefix and that nothing has gone wrong inside α earlier.)

For example, x is always substitutable for itself in any formula. If t contains no variables that occur in α, then t is substitutable for x in α.

The reader is cautioned not to be confused about the choice of words. Even if t is not substitutable for x in α, still α_t^x is obtained from α by replacing x wherever it occurs free by t. Thus in forming α_t^x, we carry out the indicated substitution even if a prudent person would think it unwise to do so.

Axiom group 2 consists of all generalizations of formulas of the form

$$\forall x\, \alpha \longrightarrow \alpha_t^x,$$

where the term t is substitutable for the variable x in the formula α. For example,

$$\forall v_3(\forall v_1(A v_1 \longrightarrow \forall v_2 A v_2) \longrightarrow (A v_2 \longrightarrow \forall v_2 A v_2))$$

is in axiom group 2. Here x is v_1, α is $A v_1 \longrightarrow \forall v_2 A v_2$, and t is v_2. On the other hand,

$$\forall v_1 \forall v_2 B v_1 v_2 \longrightarrow \forall v_2 B v_2 v_2$$

is *not* in axiom group 2, since v_2 is not substitutable for v_1 in $\forall v_2 B v_1 v_2$.

Tautologies

Axiom group 1 consists of generalizations of formulas to be called *tautologies*. These are the wffs obtainable from tautologies of sentential logic (having only the connectives $\neg$ and $\longrightarrow$) by replacing each sentence symbol by a wff of the first-order language. For example,

$$\forall x[(\forall y \neg P y \longrightarrow \neg P x) \longrightarrow (P x \longrightarrow \neg \forall y \neg P y)]$$

belongs to axiom group 1. It is a generalization of the formula in square brackets, which is obtained from a contraposition tautology

$$(\mathbf{A} \longrightarrow \neg \mathbf{B}) \longrightarrow (\mathbf{B} \longrightarrow \neg \mathbf{A})$$

by replacing $\mathbf{A}$ by $\forall y \neg P y$ and $\mathbf{B}$ by $P x$.

There is another, more direct, way of looking at axiom group 1. Divide the wffs into two groups:

1. The *prime* formulas are the atomic formulas and those of the form $\forall x\, \alpha$.

2. The nonprime formulas are the others, i.e., those of the form $\neg \alpha$ or $\alpha \longrightarrow \beta$.

Thus any formula is built up from prime formulas by the operations $\mathcal{E}_\neg$ and $\mathcal{E}_\rightarrow$. Now go back to sentential logic, but take the sentence symbols to be the prime formulas of our first-order language. Then any tautology of sentential logic (that uses only the connectives $\neg$, $\longrightarrow$) is in axiom group 1. There is no need to *replace* sentence symbols here by first-order wffs; they already *are* first-order wffs. Conversely, anything in axiom group 1 is a generalization of a tautology of sentential logic. (The proof of this uses Exercise 8 of Section 1.2.)

EXAMPLE, revisited.

$$(\forall y \neg P y \longrightarrow \neg P x) \longrightarrow (P x \longrightarrow \neg \forall y \neg P y).$$

This has two sentence symbols (prime formulas), $\forall\, y \neg Py$ and Px. So its truth table has four lines:

$$(\forall\, y \neg Py \rightarrow \neg Px) \rightarrow (Px \rightarrow \neg \forall\, y \neg Py)$$

T		F	F T	**T**	T	F	F T	
T		T	T F	**T**	F	T	F T	
F		T	F T	**T**	T	T	T F	
F		T	T F	**T**	F	T	T F	

From the table we see that it is indeed a tautology.

On the other hand, neither $\forall\, x(Px \rightarrow Px)$ nor $\forall\, x\, Px \rightarrow Px$ is a tautology.

One remark: We have not assumed that our first-order language has only countably many formulas. So we are potentially employing an extension of Chapter 1 to the case of an uncountable set of sentence symbols.

A second remark: Taking *all* tautologies as logical axioms is overkill. We could get by with much less, at the expense of lengthening deductions. On the one hand, the tautologies form a nice decidable set (the decidability will be important for the enumerability theorem in Section 2.5). On the other hand, there is no known *fast* decision procedure for tautologies, as noted in Section 1.7. One option would be to cut axiom group 1 down to a set of tautologies for which we do know decision procedures that are fast (the technical term is "polynomial-time decidable"). The other tautologies would then be obtainable from these by use of modus ponens.

A third remark: Now that first-order formulas are also wffs of sentential logic, we can apply concepts from both Chapters 1 and 2 to them. If Γ tautologically implies φ, then it follows that Γ also logically implies φ. (See Exercise 3.) But the converse fails. For example, $\forall\, x\, Px$ logically implies Pc. But $\forall\, x\, Px$ does *not* tautologically imply Pc, as $\forall\, x\, Px$ and Pc are two different sentence symbols.

THEOREM 24B $\Gamma \vdash \varphi$ iff $\Gamma \cup \Lambda$ tautologically implies φ.

PROOF.   ($\Rightarrow$): This depends on the obvious fact that $\{\alpha, \alpha \rightarrow \beta\}$ tautologically implies β. Suppose that we have a truth assignment v that satisfies every member of $\Gamma \cup \Lambda$. By induction we can see that v satisfies any theorem of Γ. The inductive step uses exactly the above-mentioned obvious fact.

($\Leftarrow$): Assume that $\Gamma \cup \Lambda$ tautologically implies φ. Then by the corollary to the compactness theorem (for sentential logic), there is a finite subset $\{\gamma_1, \ldots, \gamma_m, \lambda_1, \ldots, \lambda_n\}$ that tautologically implies φ. Consequently,

$$\gamma_1 \rightarrow \cdots \rightarrow \gamma_m \rightarrow \lambda_1 \rightarrow \cdots \lambda_n \rightarrow \varphi$$

is a tautology (cf. Exercise 4 of Section 1.2) and hence is in Λ. By applying modus ponens $m + n$ times to this tautology and to $\{\gamma_1, \ldots, \gamma_m, \lambda_1, \ldots, \lambda_n\}$ we obtain φ.   ⊣

(The above proof is related to Exercise 7 in Section 1.7. It uses sentential compactness for a possibly uncountable language.)

Deductions and Metatheorems

We now have completed the description of the set Λ of logical axioms. The set of theorems of a set Γ is the set built up from $\Gamma \cup \Lambda$ by modus ponens. For example,

$$\vdash Px \rightarrow \exists y\, Py.$$

(Here $\Gamma = \varnothing$; we write "$\vdash \alpha$" in place of "$\varnothing \vdash \alpha$.") The formula $Px \rightarrow \exists y\, Py$ can be obtained by applying modus ponens (once) to two members of Λ, as displayed by the pedigree tree:

$$Px \rightarrow \neg\forall y \neg Py$$

$$\forall y\, \neg Py \rightarrow \neg Px$$
$$\text{(in axiom group 2)}$$

$$(\forall y\, \neg Py \rightarrow \neg Px) \rightarrow (Px \rightarrow \neg\forall y \neg Py)$$
$$\text{(in axiom group 1)}$$

We get a deduction of $Px \rightarrow \exists y\, Py$ (from $\varnothing$) by compressing this tree into a linear three-element sequence.

As a second example, we can obtain a generalization of the formula in the first example:

$$\vdash \forall x (Px \rightarrow \exists y\, Py).$$

This fact is evidenced by the following tree, which displays the construction of $\forall x (Px \rightarrow \exists y\, Py)$ from Λ by modus ponens:

$$\forall x (Px \rightarrow \neg\forall y \neg Py)$$

$$\forall x (\forall y\, \neg Py \rightarrow \neg Px)$$
$$\text{(in axiom group 2)}$$

$$\forall x (\forall y\, \neg Py \rightarrow \neg Px) \rightarrow \forall x (Px \rightarrow \neg\forall y \neg Py)$$

$$\text{(a member of axiom group 3)}$$

$$\forall x [(\forall y\, \neg Py \rightarrow \neg Px) \rightarrow (Px \rightarrow \neg\forall y \neg Py)]$$
$$\text{(in axiom group 1)}$$

Again we can compress the tree into a deduction.

In these examples the pedigree trees may seem to have been pulled out of the air. But we will shortly develop techniques for generating such trees in a somewhat systematic manner. These techniques will rely heavily on the generalization theorem and the deduction theorem below.

Notice that we use the word "theorem" on two different levels. We say that α is a theorem of Γ if $\Gamma \vdash \alpha$. We also make numerous statements in English, each called a theorem, such as the one below. It seems unlikely that any confusion will arise. The English statements could have been labeled *metatheorems* to emphasize that they are results about deductions and theorems.

The generalization theorem reflects our informal feeling that if we can prove __ x __ without any special assumptions about x, we then are entitled to say that "since x was arbitrary, we have $\forall x$ __ x __."

> **GENERALIZATION THEOREM**   If $\Gamma \vdash \varphi$ and x do not occur free in any
> formula in Γ, then $\Gamma \vdash \forall x\, \varphi$.

> PROOF.   Consider a fixed set Γ and a variable x not free in Γ. We
> will show by induction that for any theorem φ of Γ, we have
> $\Gamma \vdash \forall x\, \varphi$. For this it suffices (by the induction principle) to show
> that the set
>
> $$\{\varphi \mid \Gamma \vdash \forall x\, \varphi\}$$
>
> includes $\Gamma \cup \Lambda$ and is closed under modus ponens. Notice that x
> *can* occur free in φ.
>
> Case 1: φ is a logical axiom. Then $\forall x\, \varphi$ is also a logical axiom.
> And so $\Gamma \vdash \forall x\, \varphi$.
>   Case 2: $\varphi \in \Gamma$. Then x does not occur free in φ. Hence
>
> $$\varphi \to \forall x\, \varphi$$
>
> is in axiom group 4. Consequently, $\Gamma \vdash \forall x\, \varphi$, as is evidenced by
> the tree:

$$\forall x\, \varphi$$

$$\varphi \qquad \varphi \to \forall x\, \varphi$$
$$\text{(in } \Gamma) \qquad \text{(in axiom group 4)}$$

> Case 3: φ is obtained by modus ponens from ψ and $\psi \to$
> φ. Then by inductive hypothesis we have $\Gamma \vdash \forall x\, \psi$ and $\Gamma \vdash$
> $\forall x\, (\psi \to \varphi)$. This is just the situation in which axiom group 3 is
> useful. That $\Gamma \vdash \forall x\, \varphi$ is evidenced by the tree:

$$\forall x\,\varphi$$

$$\forall x\,\psi \qquad \forall x\,\psi \rightarrow \forall x\,\varphi$$

$$\forall x(\psi \rightarrow \varphi) \qquad \forall x(\psi \rightarrow \varphi) \rightarrow (\forall x\,\psi \rightarrow \forall x\,\varphi)$$
(in axiom group 3)
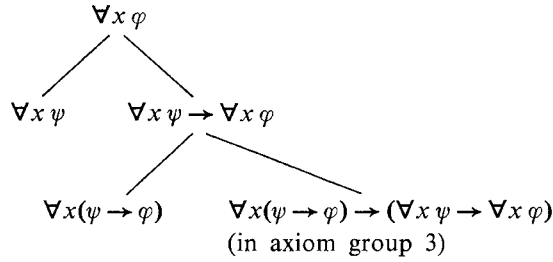
So by induction $\Gamma \vdash \forall x\,\varphi$ for every theorem φ of Γ. ⊣

(The sole reasons for having axiom groups 3 and 4 are indicated by the above proof.)

The restriction that x not occur free in Γ is essential. For example, $Px \not\models \forall x\,Px$, and so by the soundness theorem to appear in Section 2.5, $Px \not\vdash \forall x\,Px$. On the other hand, x will in general occur free in the formula φ. For example, at the beginning of this subsection we showed first that

$$\vdash (Px \rightarrow \exists y\,Py).$$

The second example there,

$$\vdash \forall x(Px \rightarrow \exists y\,Py),$$

was obtained from the first example as in case 3 of the above proof.

EXAMPLE.　$\forall x\,\forall y\,\alpha \vdash \forall y\,\forall x\,\alpha$.

The proof of the generalization theorem actually yields somewhat more than was stated. It shows how we can, given a deduction of φ from Γ, effectively transform it to obtain a deduction of $\forall x\,\varphi$ from Γ.

LEMMA 24C (RULE T)　If $\Gamma \vdash \alpha_1, \ldots, \Gamma \vdash \alpha_n$ and $\{\alpha_1, \ldots, \alpha_n\}$ tautologically implies β, then $\Gamma \vdash \beta$.

PROOF.　$\alpha_1 \rightarrow \cdots \rightarrow \alpha_n \rightarrow \beta$ is a tautology, and hence a logical axiom. Apply modus ponens n times. ⊣

DEDUCTION THEOREM　If $\Gamma; \gamma \vdash \varphi$, then $\Gamma \vdash (\gamma \rightarrow \varphi)$.

(The converse clearly holds also; in fact, the converse is essentially the rule modus ponens.)

FIRST PROOF

$\Gamma; \gamma \vdash \varphi$　iff $(\Gamma; \gamma) \cup \Lambda$ tautologically implies φ,
　　　　　　iff $\Gamma \cup \Lambda$ tautologically implies $(\gamma \rightarrow \varphi)$,
　　　　　　iff $\Gamma \vdash (\gamma \rightarrow \varphi)$. ⊣

SECOND PROOF   The second proof does not use the compactness the-
orem of sentential logic as does the first proof. It shows in a direct
way how to transform a deduction of φ from Γ; γ to obtain a de-
duction of $(\gamma \to \varphi)$ from Γ. We show by induction that for every
theorem φ of Γ; γ the formula $(\gamma \to \varphi)$ is a theorem of Γ.

  Case 1: $\varphi = \gamma$. Then obviously $\vdash (\gamma \to \varphi)$.
  Case 2: φ is a logical axiom or a member of Γ. Then $\Gamma \vdash \varphi$.
And φ tautologically implies $(\gamma \to \varphi)$, whence by rule T we have
$\Gamma \vdash (\gamma \to \varphi)$.
  Case 3: φ is obtained by modus ponens from ψ and $\psi \to \varphi$. By
the inductive hypothesis, $\Gamma \vdash (\gamma \to \psi)$ and $\Gamma \vdash (\gamma \to (\psi \to \varphi))$.
And the set $\{\gamma \to \psi, \gamma \to (\psi \to \varphi)\}$ tautologically implies $\gamma \to \varphi$.
Thus, by rule T, $\Gamma \vdash (\gamma \to \varphi)$.

  So by induction the conclusion holds for any φ deducible from
Γ; γ. $\dashv$

COROLLARY 24D (CONTRAPOSITION) Γ; $\varphi \vdash \neg\,\psi$ iff Γ; $\psi \vdash \neg\,\varphi$.

PROOF

$$\Gamma; \varphi \vdash \neg\,\psi \;\Rightarrow\; \Gamma \vdash \varphi \to \neg\,\psi \quad \text{by the deduction theorem,}$$
$$\Rightarrow\; \Gamma \vdash \psi \to \neg\,\varphi \quad \text{by rule T,}$$
$$\Rightarrow\; \Gamma; \psi \vdash \neg\,\varphi \quad \text{by modus ponens.}$$

(In the second step we use the fact that $\varphi \to \neg\,\psi$ tautologically
implies $\psi \to \neg\,\varphi$.) By symmetry, the converse holds also. $\dashv$

Say that a set of formulas is *inconsistent* iff for some β, both β and
$\neg\,\beta$ are theorems of the set. (In this event, any formula α is a theorem
of the set, since $\beta \to \neg\,\beta \to \alpha$ is a tautology.)

COROLLARY 24E (REDUCTIO AD ABSURDUM)   If Γ; φ is inconsistent,
then $\Gamma \vdash \neg\,\varphi$.

PROOF.   From the deduction theorem we have $\Gamma \vdash (\varphi \to \beta)$ and
$\Gamma \vdash (\varphi \to \neg\,\beta)$. And $\{\varphi \to \beta, \varphi \to \neg\,\beta\}$ tautologically implies
$\neg\,\varphi$. $\dashv$

EXAMPLE. $\vdash \exists\,x\,\forall\,y\,\varphi \to \forall\,y\,\exists\,x\,\varphi$.
  There are strategic advantages to working backward.
  It suffices to show that $\exists\,x\,\forall\,y\,\varphi \vdash \forall\,y\,\exists\,x\,\varphi$, by the deduction
theorem.

It suffices to show that $\exists x \forall y \, \varphi \vdash \exists x \, \varphi$, by the generalization theorem.

It suffices to show that $\neg \forall x \, \neg \forall y \, \varphi \vdash \neg \forall x \, \neg \varphi$, as this is the same as the preceding.

It suffices to show that $\forall x \, \neg \varphi \vdash \forall x \, \neg \forall y \, \varphi$, by contraposition (and rule T).

It suffices to show that $\forall x \, \neg \varphi \vdash \neg \forall y \, \varphi$, by generalization.

It suffices to show that $\{\forall x \, \neg \varphi, \forall y \, \varphi\}$ is inconsistent, by reductio ad absurdum.

And this is easy:

1. $\forall x \, \neg \varphi \vdash \neg \varphi$ by axiom group 2 and modus ponens.
2. $\forall y \, \varphi \vdash \varphi$ for the same reason.

Lines 1 and 2 show that $\{\forall x \, \neg \varphi, \forall y \, \varphi\}$ is inconsistent.

Strategy

As the preceding example indicates, the generalization and deduction theorems (and to a smaller extent the corollaries) will be very useful in showing that certain formulas are deducible. But there is still the matter of strategy: For a given Γ and φ, where should one begin in order to show that $\Gamma \vdash \varphi$? One could, in principle, start enumerating all finite sequences of wffs until one encountered a deduction of φ from Γ. Although this would be an effective procedure (for reasonable languages) for locating a deduction if one exists, it is far too inefficient to have more than theoretical interest.

One technique is to abandon formality and to give in English a proof that the truth of Γ implies the truth of φ. Then the proof in English can be formalized into a legal deduction. (In the coming pages we will see techniques for carrying out such a formalization in a reasonably natural way.)

There are also useful methods based solely on the syntactical form of φ. Assume then that φ is indeed deducible from Γ but that you are seeking a proof of this fact. There are several cases:

1. Suppose that φ is $(\psi \rightarrow \theta)$. Then it will suffice to show that $\Gamma; \psi \vdash \theta$ (and this will always be possible).

2. Suppose that φ is $\forall x \, \psi$. If x does not occur free in Γ, then it will suffice to show that $\Gamma \vdash \psi$. (Even if x should occur free in Γ, the difficulty can be circumvented. There will always be a variable y such that $\Gamma \vdash \forall y \, \psi_y^x$ and $\forall y \, \psi_y^x \vdash \forall x \, \psi$. See the re-replacement lemma, Exercise 9.)

3. Finally, suppose that φ is the negation of another formula.

3a. If φ is $\neg(\psi \rightarrow \theta)$, then it will suffice to show that $\Gamma \vdash \psi$ and $\Gamma \vdash \neg \theta$ (by rule T). And this will always be possible.

3b. If φ is $\neg \neg \psi$, then of course it will suffice to show that $\Gamma \vdash \psi$.

3c. The remaining case is where φ is $\neg \forall x \, \psi$. It would suffice to show that $\Gamma \vdash \neg \psi_t^x$, where t is some term substitutable for x in ψ. (Why?) Unfortunately this is not always possible. There are cases in which

$$\Gamma \vdash \neg \forall x \, \psi,$$

and yet for every term t,

$$\Gamma \nvdash \neg \psi_t^x.$$

(One such example is $\Gamma = \varnothing$, $\psi = \neg (P x \rightarrow \forall y \, P y)$.) Contraposition is handy here;

$$\Gamma; \alpha \vdash \neg \forall x \, \psi$$

iff

$$\Gamma; \forall x \, \psi \vdash \neg \alpha.$$

(A variation on this is: $\Gamma; \forall y \, \alpha \vdash \neg \forall x \, \psi$ if $\Gamma; \forall x \, \psi \vdash \neg \alpha$.) If all else fails, one can try reductio ad absurdum.

EXAMPLE (Q2A).   If x does not occur free in α, then

$$\vdash (\alpha \rightarrow \forall x \, \beta) \leftrightarrow \forall x (\alpha \rightarrow \beta).$$

To prove this, it suffices (by rule T) to show, that

$$\vdash (\alpha \rightarrow \forall x \, \beta) \rightarrow \forall x (\alpha \rightarrow \beta)$$

and

$$\vdash \forall x (\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \forall x \, \beta).$$

For the first of these, it suffices (by the deduction and generalization theorems) to show that

$$\{(\alpha \rightarrow \forall x \, \beta), \alpha\} \vdash \beta.$$

But this is easy; $\forall x \, \beta \rightarrow \beta$ is an axiom.
To obtain the converse,

$$\vdash \forall x (\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \forall x \, \beta),$$

it suffices (by the deduction and generalization theorems) to show that

$$\{\forall x (\alpha \rightarrow \beta), \alpha\} \vdash \beta.$$

This again is easy.

In the above example we can replace α by $\neg \alpha$, β by $\neg \beta$, and use the contraposition tautology (among other things) to obtain the related fact:

(Q3B).   If x does not occur free in α, then

$$\vdash (\exists x\, \beta \rightarrow \alpha) \leftrightarrow \forall x(\beta \rightarrow \alpha).$$

The reader might want to convince him- or herself that the above formula *is* valid.

Frequently an abbreviated style is useful in writing down a proof of deducibility, as in the following example.

EXAMPLE (EQ2). $\forall x\, \forall\, y(x = y \rightarrow y = x)$.

PROOF

    1. $\vdash x = y \rightarrow x = x \rightarrow y = x$. Ax 6.
    2. $\vdash x = x$. Ax 5.
    3. $\vdash x = y \rightarrow y = x$. 1, 2; T.
    4. $\vdash \forall x\, \forall\, y(x = y \rightarrow y = x)$. 3; gen^2. $\dashv$

In line 1, "Ax 6" means that the formula belongs to axiom group 6. In line 3, "1, 2; T" means that this line is obtained from lines 1 and 2 by rule T. In line 4, "3; gen^2" means that the generalization theorem can be applied twice to line 3 to yield line 4. In the same spirit we write "MP," "ded," and "RAA" to refer to modus ponens, the deduction theorem, and reductio ad absurdum, respectively.

It must be emphasized that the four numbered lines above do not constitute a deduction of $\forall x\, \forall\, y(x = y \rightarrow y = x)$. Instead they form a proof (in the meta-language we continue, with little justification, to call English) that such a deduction exists. The shortest deduction of $\forall x\, \forall\, y(x = y \rightarrow y = x)$ known to the author is a sequence of 17 formulas.

EXAMPLE. $\vdash x = y \rightarrow \forall z\, Pxz \rightarrow \forall z\, Pyz$.

PROOF

    1. $\vdash x = y \rightarrow Pxz \rightarrow Pyz$. Ax 6.
    2. $\vdash \forall z\, Pxz \rightarrow Pxz$. Ax 2.
    3. $\vdash x = y \rightarrow \forall z\, Pxz \rightarrow Pyz$. 1, 2; T.
    4. $\{x = y, \forall z\, Pxz\} \vdash Pyz$. 3; MP^2.
    5. $\{x = y, \forall z\, Pxz\} \vdash \forall z\, Pyz$. 4; gen.
    6. $\vdash x = y \rightarrow \forall z\, Pxz \rightarrow \forall z\, Pyz$. 5; ded^2. $\dashv$

EXAMPLE (EQ5).   Let f be a two-place function symbol. Then

$$\vdash \forall x_1\, \forall x_2\, \forall\, y_1\, \forall\, y_2(x_1 = y_1 \rightarrow x_2 = y_2 \rightarrow f x_1 x_2 = f y_1 y_2).$$

PROOF.   Two members of axiom group 6 are

$$x_1 = y_1 \rightarrow f x_1 x_2 = f x_1 x_2 \rightarrow f x_1 x_2 = f y_1 x_2,$$
$$x_2 = y_2 \rightarrow f x_1 x_2 = f y_1 x_2 \rightarrow f x_1 x_2 = f y_1 y_2.$$

From $\forall x \; x = x$ (in axiom group 5) we deduce

$$f x_1 x_2 = f x_1 x_2.$$

The three displayed formulas tautologically imply

$$x_1 = y_1 \rightarrow x_2 = y_2 \rightarrow f x_1 x_2 = f y_1 y_2. \qquad \dashv$$

EXAMPLE

(a) $\{\forall x (P x \rightarrow Q x), \forall z \; P z\} \vdash Q c$. It is not hard to show that such a deduction exists. The deduction itself consists of seven formulas.

(b) $\{\forall x (P x \rightarrow Q x), \forall z \; P z\} \vdash Q y$. This is just like (a). The point we are interested in here is that we can use the *same* seven-element deduction, with c replaced throughout by y.

(c) $\{\forall x (P x \rightarrow Q x), \forall z \; P z\} \vdash \forall y \; Q y$. This follows from (b) by the generalization theorem.

(d) $\{\forall x (P x \rightarrow Q x), \forall z \; P z\} \vdash \forall x \; Q x$. This follows from (c) by use of the fact that $\forall y \; Q y \vdash \forall x \; Q x$.

Parts (a) and (b) of the foregoing example illustrate a sort of inter-changeability of constant symbols with free variables. This interchange-ability is the basis for the following variation on the generalization theorem, for which part (c) is an example. Part (d) is covered by Corollary 24G. φ_y^c is, of course, the result of replacing c by y in φ.

THEOREM 24F (GENERALIZATION ON CONSTANTS)   Assume that $\Gamma \vdash \varphi$ and that c is a constant symbol that does not occur in Γ. Then there is a variable y (which does not occur in φ) such that $\Gamma \vdash \forall y \; \varphi_y^c$. Furthermore, there is a deduction of $\forall y \; \varphi_y^c$ from Γ in which c does not occur.

PROOF.   Let $\langle \alpha_0, \ldots, \alpha_n \rangle$ be a deduction of φ from Γ. (Thus $\alpha_n = \varphi$.) Let y be the first variable that does not occur in any of the α_i's. We claim that

$$\langle (\alpha_0)_y^c, \ldots, (\alpha_n)_y^c \rangle \qquad\qquad (*)$$

is a deduction from Γ of φ_y^c. So we must check that each $(\alpha_k)_y^c$ is in $\Gamma \cup \Lambda$ or is obtained from earlier formulas by modus ponens.

Case 1: $\alpha_k \in \Gamma$. Then c does not occur in α_k. So $(\alpha_k)_y^c = \alpha_k$, which is in Γ.

Case 2: α_k is a logical axiom. Then $(\alpha_k)_y^c$ is also a logical axiom. (Read the list of logical axioms and note that introducing a new variable will transform a logical axiom into another one.)

Case 3: α_k is obtained by modus ponens from α_i and α_j (which is $(\alpha_i \rightarrow \alpha_k)$) for i, j less than k. Then $(\alpha_j)_y^c = ((\alpha_i)_y^c \rightarrow (\alpha_k)_y^c)$. So $(\alpha_k)_y^c$ is obtained by modus ponens from $(\alpha_i)_y^c$ and $(\alpha_j)_y^c$.

This completes the proof that $(*)$ above is a deduction of φ_y^c. Let Φ be the finite subset of Γ that is used in $(*)$. Thus $(*)$ is a deduction of φ_y^c from Φ, and y does not occur in Φ. So by the generalization theorem, $\Phi \vdash \forall y \, \varphi_y^c$. Furthermore, there is a deduction of $\forall y \, \varphi_y^c$ from Φ in which c does not appear. (For the proof to the generalization theorem did not add any new symbols to a deduction.) This is also a deduction from Γ of $\forall y \, \varphi_y^c$. $\dashv$

We will sometimes want to apply this theorem in circumstances in which not just any variable will do. In the following version, there is a variable x selected in advance.

COROLLARY 24G   Assume that $\Gamma \vdash \varphi_c^x$, where the constant symbol c does not occur in Γ or in φ. Then $\Gamma \vdash \forall x \, \varphi$, and there is a deduction of $\forall x \, \varphi$ from Γ in which c does not occur.

PROOF.   By the above theorem we have a deduction (without c) from Γ of $\forall y((\varphi_c^x)_y^c)$, where y does not occur in φ_c^x. But since c does not occur in φ,

$$\left(\varphi_c^x\right)_y^c = \varphi_y^x.$$

It remains to show that $\forall y \, \varphi_y^x \vdash \forall x \, \varphi$. We can easily do this if we know that

$$(\forall y \, \varphi_y^x) \to \varphi$$

is an axiom. That is, x must be substitutable for y in φ_y^x, and $(\varphi_y^x)_x^y$ must be φ. This is reasonably clear; for details see the re-replacement lemma (Exercise 9). $\dashv$

COROLLARY 24H (RULE EI)   Assume that the constant symbol c does not occur in φ, ψ, or Γ, and that

$$\Gamma; \varphi_c^x \vdash \psi.$$

Then

$$\Gamma; \exists x \, \varphi \vdash \psi$$

and there is a deduction of ψ from $\Gamma; \exists x \, \varphi$ in which c does not occur.

PROOF.   By contraposition we have

$$\Gamma; \neg \psi \vdash \neg \varphi_c^x.$$

So by the preceding corollary we obtain

$$\Gamma; \neg \psi \vdash \forall x \, \neg \varphi.$$

Applying contraposition again, we have the desired result. $\dashv$

"EI" stands for "existential instantiation," a bit of traditional terminology.

We will not have occasion to use rule EI in any of our proofs, but it may be handy in exercises. It is the formal counterpart to the reasoning: "We know there is an x such that __ x __. So call it c. Now from __ c __ we can prove ψ." But notice that rule EI does *not* claim that $\exists x \, \varphi \vdash \varphi_c^x$, which is in fact usually false.

> **EXAMPLE**, revisited. $\vdash \exists x \, \forall y \, \varphi \rightarrow \forall y \, \exists x \, \varphi$.
>
> By the deduction theorem, it suffices to show that
>
> $$\exists x \, \forall y \, \varphi \vdash \forall y \, \exists x \, \varphi.$$
>
> By rule EI it suffices to show that
>
> $$\forall y \, \varphi_c^x \vdash \forall y \, \exists x \, \varphi,$$
>
> where c is new to the language. By the generalization theorem it suffices to show that
>
> $$\forall y \, \varphi_c^x \vdash \exists x \, \varphi.$$
>
> Since $\forall y \, \varphi_c^x \vdash \varphi_c^x$, it suffices to show that
>
> $$\varphi_c^x \vdash \exists x \, \varphi.$$
>
> By contraposition this is equivalent to
>
> $$\forall x \, \neg \varphi \vdash \neg \varphi_c^x,$$
>
> which is trivial (by axiom group 2 and modus ponens).

We can now see roughly how our particular list of logical axioms was formed. The tautologies were included to handle the sentential connective symbols. (We could economize considerably at this point by using only some of the tautologies.) Axiom group 2 reflects the intended meaning of the quantifier symbol. Then in order to be able to prove the generalization theorem we added axiom groups 3 and 4 and arranged for generalizations of axioms to be axioms.

Axiom groups 5 and 6 will turn out to be just enough to prove the crucial properties of equality; see the subsection on equality.

As we will prove in Section 2.5, every logical axiom is a valid formula. It might seem simpler to use as logical axioms the set of *all* valid formulas. But there are two (related) objections to doing this. For one, the concept of validity was defined *semantically*. That is, the definition referred to possible meanings (i.e., structures) for the language and to the concept of truth in a structure. For our present purposes (e.g., proving that the validities are effectively enumerable) we need a class Λ with a finitary, *syntactical* definition. That is, the definition of Λ involves only matters concerning the arrangement of the symbols in the logical axioms; there is no reference to matters of truth in structures. A second

objection to the inclusion of all valid formulas as axioms is that we prefer a decidable set Λ, and the set of validities fails to be decidable.

Alphabetic Variants

Often when we are discussing a formula such as

$$\forall x (x \neq 0 \rightarrow \exists y \, x = \mathbf{S} y)$$

we are not interested in the particular choice of the variables x and y. We want $\langle x, y \rangle$ to be a pair of distinct variables, but often it makes no difference whether the pair is $\langle v_4, v_9 \rangle$ or $\langle v_8, v_1 \rangle$.

But when it comes time to substitute a term t into a formula, then the choice of quantified variables can make the difference between the substitutability of t and its failure. In this subsection we will discuss what to do when substitutability fails. As will be seen, the difficulty can always be surmounted by suitably juggling the quantified variables.

For example, suppose we want to show that

$$\vdash \forall x \, \forall y \, Pxy \rightarrow \forall y \, Pyy.$$

There is the difficulty that y is not substitutable for x in $\forall y \, Pxy$, so the above sentence is not in axiom group 2. This is a nuisance resulting from an unfortunate choice of variables. For example, showing that

$$\vdash \forall x \, \forall z \, Pxz \rightarrow \forall y \, Pyy$$

involves no such difficulties. So we can solve our original problem if we know that

$$\vdash \forall x \, \forall y \, Pxy \rightarrow \forall x \, \forall z \, Pxz,$$

which, again, involves no difficulties.

This slightly circuitous strategy (of interpolating $\forall x \, \forall z \, Pxz$ between $\forall x \, \forall y \, Pxy$ and $\forall y \, Pyy$) is typical of a certain class of problems. Say that we desire to substitute a term t for x in a wff φ. If t is, in fact, not so substitutable, then we replace $\forall x \, \varphi$ by $\forall x \, \varphi'$, where t is substitutable for x in φ'. In the above example φ is $\forall y \, Pxy$ and φ' is $\forall z \, Pxz$. In general φ' will differ from φ only in the choice of quantified variables. But φ' must be formed in a reasonable way so as to be logically equivalent to φ. For example, it would be unreasonable to replace $\forall y \, Pxy$ by $\forall x \, Pxx$, or $\forall y \, \forall z \, Qxyz$ by $\forall z \, \forall z \, Qxzz$.

THEOREM 24I (EXISTENCE OF ALPHABETIC VARIANTS)   Let φ be a formula, t a term, and x a variable. Then we can find a formula φ' (which differs from φ only in the choice of quantified variables) such that

(a) $\varphi \vdash \varphi'$ and $\varphi' \vdash \varphi$;
(b) t is substitutable for x in φ'.

PROOF. We consider fixed t and x, and construct φ' by recursion on φ. The first cases are simple: For atomic φ we take $\varphi' = \varphi$, and then $(\neg\,\varphi)' = (\neg\,\varphi')$, $(\varphi \rightarrow \psi)' = (\varphi' \rightarrow \psi')$. But now consider the choice of $(\forall\, y\, \varphi)'$.

If y does not occur in t, or if $y = x$, then we can just take $(\forall\, y\, \varphi)' = \forall\, y\, \varphi'$. But for the general case we must change the variable.

Choose a variable z that does not occur in φ' or t or x. Then define $(\forall\, y\, \varphi)' = \forall\, z(\varphi')^y_z$. To verify that (b) holds, we note that z does not occur in t and t is substitutable for x in φ' (by the inductive hypothesis). Hence (since $x \neq z$) t is also substitutable for x in $(\varphi')^y_z$. To verify that (a) holds, we calculate:

$$\varphi \vdash \varphi' \qquad\qquad \text{by the inductive hypothesis;}$$
$$\therefore \forall\, y\, \varphi \vdash \forall\, y\, \varphi'.$$
$$\forall\, y\, \varphi' \vdash (\varphi')^y_z \qquad \text{since } z \text{ does not occur in } \varphi';$$
$$\therefore \forall\, y\, \varphi' \vdash \forall\, z(\varphi')^y_z \quad \text{by generalization;}$$
$$\therefore \forall\, y\, \varphi \vdash \forall\, z(\varphi')^y_z.$$

In the other direction,

$$\forall\, z(\varphi')^y_z \vdash ((\varphi')^y_z)^z_y, \quad \text{which is } \varphi' \text{ by Exercise 9;}$$
$$\varphi' \vdash \varphi; \qquad\qquad \text{by the inductive hypothesis;}$$
$$\therefore \forall\, z(\varphi')^y_z \vdash \varphi$$
$$\therefore \forall\, z(\varphi')^y_z \vdash \forall\, y\, \varphi \quad \text{by generalization.}$$

The last step uses the fact that y does not occur free in $(\varphi')^y_z$ unless $y = z$, and so does not occur free in $\forall\, z(\varphi')^y_z$ in any case.   ⊣

The formulas φ' constructed as in the proof of this theorem will be called *alphabetic variants* of φ. The moral of the theorem is: One should not be daunted by failure of substitutability; the right alphabetic variant will avoid the difficulty.

Equality

We list here (assuming that our language includes $=$) the facts about equality that will be needed in the next section. First, the relation defined by $v_1 = v_2$ is reflexive, symmetric, and transitive (i.e., is an equivalence relation):

$$\text{Eq1:} \quad \vdash \forall\, x\, x = x.$$

PROOF. Axiom group 5.   ⊣

$$\text{Eq2:} \quad \vdash \forall\, x\, \forall\, y(x = y \rightarrow y = x).$$

PROOF.   Page 122.   ⊣

Eq3:   ⊢ ∀ x ∀ y ∀ z(x = y → y = z → x = z).

PROOF.   Exercise 11.   ⊣

In addition, we will need to know that equality is compatible with the predicate and function symbols:

Eq4   (for a two-place predicate symbol P):

$$⊢ ∀ x_1 ∀ x_2 ∀ y_1 ∀ y_2(x_1 = y_1 → x_2 = y_2 → Px_1x_2 → Py_1y_2).$$

Similarly for n-place predicate symbols.

PROOF.   It suffices to show that

$$\{x_1 = y_1, x_2 = y_2, Px_1x_2\} ⊢ Py_1y_2.$$

This is obtained by application of modus ponens to the two members of axiom group 6:

$$x_1 = y_1 → Px_1x_2 → Py_1x_2,$$
$$x_2 = y_2 → Py_1x_2 → Py_1y_2.$$
                                                                            ⊣

Eq5 (for a two-place function symbol f):

$$⊢ ∀ x_1 ∀ x_2 ∀ y_1 ∀ y_2(x_1 = y_1 → x_2 = y_2 → fx_1x_2 = fy_1y_2).$$

Similarly for n-place function symbols.

PROOF.   Page 122.   ⊣

Final Comments

A logic book in the bootstrap tradition might well begin with this section on a deductive calculus. Such a book would first state the logical axioms and the rules of inference and would explain that they are acceptable to reasonable people. Then it would proceed to show that many formulas were deducible (or deducible from certain nonlogical axioms, such as axioms for set theory).

Our viewpoint is very different. We study, among other things, the facts about the procedure described in the preceding paragraph. And we employ in this any correct mathematical reasoning, whether or not such reasoning is known to have counterparts in the deductive calculus under study.

Figure 8 is intended to illustrate the separation between (a) the level at which we carry out our reasoning and prove our results, and (b) the level of the deductive calculus which we study.

The study, in English, of the view below:

$$\text{If } \Gamma \models \varphi, \text{ then } \Gamma \vdash \varphi.$$

$$\Gamma; \alpha \vdash \beta \wedge \neg \beta \Rightarrow \Gamma \vdash \neg \alpha$$

$$\models_{\mathfrak{R}} \forall x \neg Sx \approx 0$$

$$\models \exists x (Px \to \forall x Px)$$

Formal
language
expressions:

$$\forall x \neg Sx \approx 0$$

Axioms:

$$Py \to Py$$

$$\exists x (Px \to \forall x \, Px)$$

Deductions:

$$Qc$$

$$\forall x \, Qx$$

$$\forall x \, Qx \to Qc$$
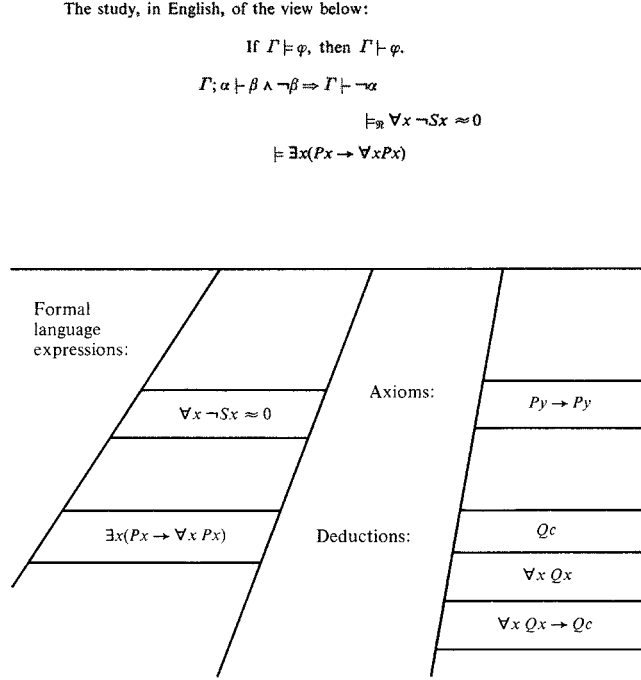
Figure 8. The meta-language above, in which we study the object
language below.

Exercises

1. For a term u, let u_t^x be the expression obtained from u by replacing
the variable x by the term t. Restate this definition without using
any form of the word "replace" or its synonyms. *Suggestion*: Use
recursion on u. (Observe that from the new definition it is clear that
u_t^x is itself a term.)

2. To which axiom groups, if any, do each of the following formulas
belong?
   (a) $[(\forall x \, Px \to \forall y \, Py) \to Pz] \to [\forall x \, Px \to (\forall y \, Py \to Pz)]$.
   (b) $\forall y[\forall x (Px \to Px) \to (Pc \to Pc)]$.
   (c) $\forall x \exists y \, Pxy \to \exists y \, Pyy$.

3. (a) Let $\mathfrak{A}$ be a structure and let $s : V \to |\mathfrak{A}|$. Define a truth assign-
ment v on the set of prime formulas by

$$v(\alpha) = T \quad \text{iff} \quad \models_{\mathfrak{A}} \alpha[s].$$

Show that for any formula (prime or not),

$$\bar{v}(\alpha) = T \quad \text{iff} \quad \models_{\mathfrak{A}} \alpha[s].$$

Remark: This result reflects the fact that $\neg$ and $\rightarrow$ were treated in Chapter 2 the same way as in Chapter 1.

(b) Conclude that if Γ tautologically implies φ, then Γ logically implies φ.

4. Give a deduction (from $\varnothing$) of $\forall x \, \varphi \rightarrow \exists x \, \varphi$. (Note that you should not merely prove that such a deduction exists. You are instead asked to write out the entire deduction.)

5. Find a function f such that if a formula φ has a deduction of length n from a set Γ, and if x does not occur free in Γ, then $\forall x \, \varphi$ has a deduction from Γ of length $f(n)$. The more slowly your function grows, the better.

6. (a) Show that if $\vdash \alpha \rightarrow \beta$, then $\vdash \forall x \, \alpha \rightarrow \forall x \, \beta$.
   (b) Show that it is not in general true that $\alpha \rightarrow \beta \models \forall x \, \alpha \rightarrow \forall x \, \beta$.

7. (a) Show that $\vdash \exists x (Px \rightarrow \forall x \, Px)$.
   (b) Show that $\{Qx, \forall y (Qy \rightarrow \forall z \, Pz)\} \vdash \forall x \, Px$.

8. (Q2b) Assume that x does not occur free in α. Show that

$$\vdash (\alpha \rightarrow \exists x \, \beta) \leftrightarrow \exists x (\alpha \rightarrow \beta).$$

Also show that, under the same assumption, we have Q3a:

$$\vdash (\forall x \, \beta \rightarrow \alpha) \leftrightarrow \exists x (\beta \rightarrow \alpha).$$

9. (Re-replacement lemma) (a) Show by example that $(\varphi_y^x)_x^y$ is not in general equal to φ. And that it is possible both for x to occur in $(\varphi_y^x)_x^y$ at a place where it does not occur in φ, and for x to occur in φ at a place where it does not occur in $(\varphi_y^x)_x^y$.
   (b) Show that if y does not occur at all in φ, then x is substitutable for y in φ_y^x and $(\varphi_y^x)_x^y = \varphi$. *Suggestion*: Use induction on φ.

10. Show that

$$\forall x \, \forall y \, Pxy \vdash \forall y \, \forall x \, Pyx.$$

11. (Eq3) Show that

$$\vdash \forall x \, \forall y \, \forall z (x = y \rightarrow y = z \rightarrow x = z).$$

12. Show that any consistent set Γ of formulas can be extended to a consistent set Δ having the property that for any formula α, either $\alpha \in \Delta$ or $(\neg \alpha) \in \Delta$. (Assume that the language is countable. Do not use the compactness theorem of sentential logic.)

13. Show that $\vdash Py \leftrightarrow \forall x (x = y \rightarrow Px)$.
    Remarks: More generally, if t is substitutable for x in φ and x does not occur in t, then

$$\vdash [\varphi_t^x \leftrightarrow \forall x (x = t \rightarrow \varphi)].$$

Thus the formula $\forall x (x = t \rightarrow \varphi)$ offers an alternative of sorts to the substitution φ_t^x.

14. Show that $\vdash (\forall x((\neg Px) \rightarrow Qx) \rightarrow \forall y((\neg Qy) \rightarrow Py))$.

15. Show that deductions (from $\varnothing$) of the following formulas exist:
    (a) $\exists x\, \alpha \vee \exists x\, \beta \leftrightarrow \exists x(\alpha \vee \beta)$.
    (b) $\forall x\, \alpha \vee \forall x\, \beta \rightarrow \forall x(\alpha \vee \beta)$.

16. Show that deductions (from $\varnothing$) of the following formulas exist:
    (a) $\exists x(\alpha \wedge \beta) \rightarrow \exists x\, \alpha \wedge \exists x\, \beta$.
    (b) $\forall x(\alpha \wedge \beta) \leftrightarrow \forall x\, \alpha \wedge \forall x\, \beta$.

17. Show that deductions (from $\varnothing$) of the following formulas exist:
    (a) $\forall x(\alpha \rightarrow \beta) \rightarrow (\exists x\, \alpha \rightarrow \exists x\, \beta)$.
    (b) $\exists x(Py \wedge Qx) \leftrightarrow Py \wedge \exists x\, Qx$.

SECTION 2.5
Soundness and Completeness Theorems

In this section we establish two major theorems: the soundness of our deductive calculus ($\Gamma \vdash \varphi \Rightarrow \Gamma \models \varphi$) and its completeness ($\Gamma \models \varphi \Rightarrow \Gamma \vdash \varphi$). We will then be able to draw a number of interesting conclusions (including the compactness and enumerability theorems). Although our deductive calculus was chosen in a somewhat arbitrary way, the significant fact is that *some* such deductive calculus is sound and complete. This should be encouraging to the "working mathematician" concerned about the existence of proofs from axioms; see the *Retrospectus* subsection of Section 2.6.

SOUNDNESS THEOREM   If $\Gamma \vdash \varphi$, then $\Gamma \models \varphi$.

The soundness theorem tells that our deductions lead only to "correct" conclusions — deductions would be rather pointless otherwise! The idea of the proof is that the logical axioms are logically implied by anything, and that modus ponens preserves logical implications.

LEMMA 25A   Every logical axiom is valid.

PROOF OF THE SOUNDNESS THEOREM, ASSUMING THE LEMMA.   We show by induction that any formula φ deducible from Γ is logically implied by Γ.

Case 1: φ is a logical axiom. Then by the lemma $\models \varphi$, so *a fortiori* $\Gamma \models \varphi$.

Case 2: $\varphi \in \Gamma$. Then clearly $\Gamma \models \varphi$.

Case 3: φ is obtained by modus ponens from ψ and $\psi \rightarrow \varphi$, where (by the inductive hypothesis) $\Gamma \models \psi$ and $\Gamma \models (\psi \rightarrow \varphi)$. It then follows at once that $\Gamma \models \varphi$.   ⊣

It remains, of course, to prove that lemma. We know from Exercise 6 of Section 2.2 that any generalization of a valid formula is valid. So it suffices to consider only logical axioms that are not themselves generalizations of other axioms. We will examine the various axiom groups in order of complexity.

Axiom group 3: See Exercise 3 of Section 2.2.
Axiom group 4: See Exercise 4 of Section 2.2.
Axiom group 5: Trivial. $\mathfrak{A}$ satisfies $x = x$ with s iff $s(x) = s(x)$, which is always true.
Axiom group 1: We know from Exercise 3 of the preceding section that if $\varnothing$ tautologically implies α, then $\varnothing \models \alpha$. And that is just what we need.
Axiom group 6 (for an example, see Exercise 5 of Section 2.2): Assume that α is atomic and α' is obtained from α by replacing x at some places by y. It suffices to show that

$$\{x = y, \alpha\} \models \alpha'.$$

So take any $\mathfrak{A}$, s such that

$$\models_{\mathfrak{A}} x = y[s], \quad \text{i.e., } s(x) = s(y).$$

Then any term t has the property that if t' is obtained from t by replacing x at some places by y, then $\bar{s}(t) = \bar{s}(t')$. This is obvious; a full proof would use induction on t.

If α is $t_1 = t_2$, then α' must be $t_1' = t_2'$, where t_i' is obtained from t_i as described.

$$
\begin{aligned}
\models_{\mathfrak{A}} \alpha[s] \quad &\text{iff} \quad \bar{s}(t_1) = \bar{s}(t_2), \\
&\text{iff} \quad \bar{s}(t_1') = \bar{s}(t_2'), \\
&\text{iff} \quad \models_{\mathfrak{A}} \alpha'[s].
\end{aligned}
$$

Similarly, if α is $Pt_1 \cdots t_n$, then α' is $Pt_1' \cdots t_n'$ and an analogous argument applies.

Finally, we come to axiom group 2. It will be helpful to consider first a simple case: we will show that $\forall x \, Px \rightarrow Pt$ is valid. Assume that

$$\models_{\mathfrak{A}} \forall x \, Px[s].$$

Then for any d in $|\mathfrak{A}|$,

$$\models_{\mathfrak{A}} Px[s(x \mid d)].$$

So in particular we may take $d = \bar{s}(t)$:

$$\models_{\mathfrak{A}} Px[s(x \mid \bar{s}(t))]. \tag{a}$$

This is equivalent (by the definition of satisfaction of atomic formulas) to

$$\bar{s}(t) \in P^{\mathfrak{A}},$$

which in turn is equivalent to

$$\models_{\mathfrak{A}} Pt[s]. \tag{b}$$

For this argument to be applicable to the nonatomic case, we need a way of passing from (a) to (b). This will be provided by the substitution lemma below, which states that

$$\models_{\mathfrak{A}} \varphi[s(x \mid \bar{s}(t))] \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi_t^x[s]$$

whenever t is substitutable for x in φ.

Consider a fixed $\mathfrak{A}$ and s. For any term u, let u_t^x be the result of replacing the variable x in u by the term t.

LEMMA 25B $\bar{s}(u_t^x) = \overline{s(x \mid \bar{s}(t))}(u)$.

This looks more complicated than it is. It asserts that a substitution can be carried out either in the term u or in s, with equivalent results. The corresponding commutative diagram is shown.
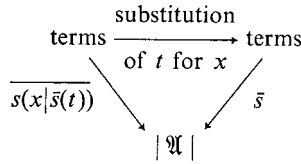
PROOF.   By induction on the term u. If u is a constant symbol or a variable other than x, then $u_t^x = u$ and the desired equation reduces to $\bar{s}(u) = \bar{s}(u)$. If $u = x$, then the equation reduces to $\bar{s}(t) = \bar{s}(t)$. The inductive step, although cumbersome to write, is mathematically trivial.   ⊣

The substitution lemma is similar in spirit; it states that a substitution can be carried out either within φ or in s, with equivalent results. For an example see Exercise 10 of Section 2.2.

SUBSTITUTION LEMMA   If the term t is substitutable for the variable x in the wff φ, then

$$\models_{\mathfrak{A}} \varphi_t^x[s] \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi[s(x \mid \bar{s}(t))].$$

PROOF.   We use induction on φ to show that the above holds for every s.

Case 1: φ is atomic. Then the conclusion follows from the preceding lemma. For example, if φ is Pu for some term u, then

$$\models_{\mathfrak{A}} Pu_t^x[s] \text{ iff } \bar{s}(u_t^x) \in P^{\mathfrak{A}},$$
$$\text{iff } \overline{s(x \mid \bar{s}(t))}(u) \in P^{\mathfrak{A}} \quad \text{by Lemma 25B,}$$
$$\text{iff } \models_{\mathfrak{A}} Pu[s(x \mid \bar{s}(t))].$$

Case 2: φ is $\neg\,\psi$ or $\psi\rightarrow\theta$. Then the conclusion for φ follows at once from the inductive hypotheses for ψ and θ.

Case 3: φ is $\forall\,y\,\psi$, and x does not occur free in φ. Then s and $s(x\,|\,\overline{s}(t))$ agree on all variables that occur free in φ. And also φ_t^x is simply φ. So the conclusion is immediate.

Case 4: φ is $\forall\,y\,\psi$, and x *does* occur free in φ. Because t is substitutable for x in φ, we know that y does not occur in t and t is substitutable for x in ψ (see the definition of "substitutable"). By the first of these,

$$\overline{s}(t) = \overline{s(y\,|\,d)}(t) \qquad (*)$$

for any d in $|\mathfrak{A}|$. Since $x\neq y$, $\varphi_t^x = \forall\,y\,\psi_t^x$.

$\models_{\mathfrak{A}}\varphi_t^x[s]$ iff for every d, $\models_{\mathfrak{A}}\psi_t^x[s(y\,|\,d)]$,
iff for every d, $\models_{\mathfrak{A}}\psi[s(y\,|\,d)(x\,|\,\overline{s}(t))]$ by the inductive hypothesis and $(*)$,
iff $\models_{\mathfrak{A}}\varphi[s(x\,|\,\overline{s}(t))]$.

So by induction the lemma holds for all φ. $\dashv$

Axiom group 2: Assume that t is substitutable for x in φ. Assume that $\mathfrak{A}$ satisfies $\forall\,x\,\varphi$ with s. We need to show that $\models_{\mathfrak{A}}\varphi_t^x[s]$. We know that for any d in $|\mathfrak{A}|$,

$$\models_{\mathfrak{A}}\varphi[s(x\,|\,d)],$$

In particular, let $d = \overline{s}(t)$:

$$\models_{\mathfrak{A}}\varphi[s(x\,|\,\overline{s}(t))],$$

So, by the substitution lemma,

$$\models_{\mathfrak{A}}\varphi_t^x[s].$$

Hence $\forall\,x\,\varphi\rightarrow\varphi_t^x$ is valid.

This completes the proof that all logical axioms are valid. And so the soundness theorem is proved.

COROLLARY 25C   If $\vdash(\varphi\leftrightarrow\psi)$, then φ and ψ are logically equivalent.

COROLLARY 25D   If φ' is an alphabetic variant of φ (see Theorem 24I), then φ and φ' are logically equivalent.

Recall that a set Γ is consistent iff there is no formula φ such that both $\Gamma\vdash\varphi$ and $\Gamma\vdash\neg\,\varphi$. Define Γ to be *satisfiable* iff there is some $\mathfrak{A}$ and s such that $\mathfrak{A}$ satisfies every member of Γ with s.

COROLLARY 25E   If Γ is satisfiable, then Γ is consistent.

This corollary is actually equivalent to the soundness theorem, as the reader is invited to verify.

The completeness theorem is the converse to the soundness theorem and is a deeper result.

COMPLETENESS THEOREM (GÖDEL, 1930)

> (a) If $\Gamma \models \varphi$, then $\Gamma \vdash \varphi$.
> (b) Any consistent set of formulas is satisfiable.

Actually parts (a) and (b) are equivalent; cf. Exercise 2. So it suffices to prove part (b). We will give a proof for a countable language; later we will indicate what alterations are needed for languages of larger cardinality. (A countable language is one with countably many symbols, or equivalently, by Theorem 0B, one with countably many wffs.)

The ideas of the proof are related to those in the proof of the compactness theorem for sentential logic. We begin with a consistent set Γ. In steps 1–3 we extend Γ to a set Δ of formulas for which

> (i) $\Gamma \subseteq \Delta$.
> (ii) Δ is consistent and is maximal in the sense that for any formula α, either $\alpha \in \Delta$ or $(\neg \alpha) \in \Delta$.
> (iii) For any formula φ and variable x, there is a constant c such that

$$(\neg \forall x \, \varphi \rightarrow \neg \varphi_c^x) \in \Delta.$$

Then in step 4 we form a structure $\mathfrak{A}$ in which members of Γ not containing $=$ can be satisfied. $|\mathfrak{A}|$ is the set of terms, and for a predicate symbol P,

$$\langle t_1, \ldots, t_n \rangle \in P^{\mathfrak{A}} \quad \text{iff} \quad Pt_1 \cdots t_n \in \Delta.$$

Finally, in steps 5 and 6 we change $\mathfrak{A}$ to accommodate formulas containing the equality symbol.

It is suggested that on a first reading the details which are provided for most of the steps be omitted. Once the outline is clearly in mind, the entire proof should be read. (The nondetails are marked with a stripe in the left margin.)

> PROOF.   Let Γ be a consistent set of wffs in a countable language.
> $\dashv$
>
> STEP 1:   Expand the language by adding a countably infinite set of new constant symbols. Then Γ remains consistent as a set of wffs in the new language.

Details: If not then for some β, there is a deduction (in the expanded language) of $(\beta \wedge \neg \beta)$ from Γ. This deduction contains only finitely many of the new constant symbols. By the theorem for generalization on constants (Theorem 24F), each can be replaced by a variable. We then have a deduction (in the original language) of $(\beta' \wedge \neg \beta')$ from Γ. This contradicts our assumption that Γ was consistent.

STEP 2:   For each wff φ (in the new language) and each variable x, we want to add to Γ the wff

$$\neg \forall x\, \varphi \rightarrow \neg \varphi_c^x,$$

where c is one of the new constant symbols. (The idea is that c volunteers to name a counterexample to φ, if there is any.) We can do this in such a way that Γ together with the set Θ of all the added wffs is still a consistent set.

Details: Adopt a fixed enumeration of the pairs $\langle \varphi, x \rangle$, where φ is a wff (of the expanded language) and x is a variable:

$$\langle \varphi_1, x_1 \rangle, \langle \varphi_2, x_2 \rangle, \langle \varphi_3, x_3 \rangle, \ldots.$$

This is possible since the language is countable. Let θ_1 be

$$\neg \forall x_1 \varphi_1 \rightarrow \neg \varphi_1{}_{c_1}^{x_1},$$

where c_1 is the first of the new constant symbols not occurring in φ_1. Then go on to $\langle \varphi_2, x_2 \rangle$ and define θ_2. In general, θ_n is

$$\neg \forall x_n \varphi_n \rightarrow \neg \varphi_n{}_{c_n}^{x_n},$$

where c_n is the first of the new constant symbols not occurring in φ_n or in θ_k for any $k < n$.

Let Θ be the set $\{\theta_1, \theta_2, \ldots\}$. We claim that $\Gamma \cup \Theta$ is consistent. If not, then (because deductions are finite) for some $m \geq 0$,

$$\Gamma \cup \{\theta_1, \ldots, \theta_m, \theta_{m+1}\}$$

is inconsistent. Take the least such m. Then by RAA

$$\Gamma \cup \{\theta_1, \ldots, \theta_m\} \vdash \neg \theta_{m+1}.$$

Now θ_{m+1} is

$$\neg \forall x\, \varphi \rightarrow \neg \varphi_c^x$$

for some x, φ, and c. So by rule T, we obtain the two facts:

$$\begin{aligned}
\Gamma \cup \{\theta_1, \ldots \theta_m\} &\vdash \neg \forall x\, \varphi, \\
\Gamma \cup \{\theta_1, \ldots \theta_m\} &\vdash \varphi_c^x.
\end{aligned} \tag{$*$}$$

Since c does not appear in any formula on the left side, we can apply the Corollary 24G to the second of these, obtaining

$$\Gamma \cup \{\theta_1, \ldots, \theta_m\} \vdash \forall x\, \varphi.$$

This and ($*$) contradict the leastness of m (or the consistency of Γ, if $m = 0$).

STEP 3:   We now extend the consistent set $\Gamma \cup \Theta$ to a consistent set Δ which is maximal in the sense that for any wff φ either $\varphi \in \Delta$ or $(\neg \varphi) \in \Delta$.

Details: We can imitate the proof used at the analogous place in the proof of sentential compactness in Section 1.7. Or we can argue as follows: Let Λ be the set of logical axioms for the expanded language. Since $\Gamma \cup \Theta$ is consistent, there is no formula β such that $\Gamma \cup \Theta \cup \Lambda$ tautologically implies both β and $\neg \beta$. (This is by Theorem 24B; the compactness theorem of sentential logic is used here.) Hence there is a truth assignment v for the set of all prime formulas that satisfies $\Gamma \cup \Theta \cup \Lambda$. Let

$$\Delta = \{\varphi \mid \overline{v}(\varphi) = T\}.$$

Clearly for any φ either $\varphi \in \Delta$ or $(\neg \varphi) \in \Delta$ but not both. Also we have

$$\begin{aligned}
\Delta \vdash \varphi \Rightarrow{}& \Delta \text{ tautologically implies } \varphi \quad \text{(since } \Lambda \subseteq \Delta), \\
\Rightarrow{}& \overline{v}(\varphi) = T \qquad\qquad\qquad\quad \text{since } v \text{ satisfies } \Delta, \\
\Rightarrow{}& \varphi \in \Delta.
\end{aligned}$$

Consequently, Δ is consistent, lest both φ and $(\neg \varphi)$ belong to Δ.

Actually, regardless of how Δ is constructed, it must be deductively closed. That is,

$$\begin{aligned}
\Delta \vdash \varphi \Rightarrow{}& \Delta \nvdash \neg \varphi \quad \text{by consistency,} \\
\Rightarrow{}& (\neg \varphi) \notin \Delta, \\
\Rightarrow{}& \varphi \in \Delta \qquad \text{by maximality.}
\end{aligned}$$

STEP 4:   We now make from Δ a structure $\mathfrak{A}$ for the new language, but with the equality symbol (if any) replaced by a new two-place predicate symbol E. $\mathfrak{A}$ will not itself be the structure in which Γ will be satisfied but will be a preliminary structure.

(a) $|\mathfrak{A}|$ = the set of all terms of the new language.
(b) Define the binary relation $E^{\mathfrak{A}}$ by

$$\langle u, t \rangle \in E^{\mathfrak{A}} \quad \text{iff} \quad \text{the formula } u = t \text{ belongs to } \Delta.$$

(c) For each n-place predicate parameter P, define the n-ary relation $P^{\mathfrak{A}}$ by

$$\langle t_1, \ldots, t_n \rangle \in P^{\mathfrak{A}} \quad \text{iff} \quad P t_1 \cdots t_n \in \Delta.$$

(d) For each n-place function symbol f, let $f^{\mathfrak{A}}$ be the function defined by

$$f^{\mathfrak{A}}(t_1, \ldots, t_n) = f t_1 \cdots t_n.$$

This includes the $n = 0$ case; for a constant symbol c we take $c^{\mathfrak{A}} = c$. Define also a function $s : V \to |\mathfrak{A}|$, namely the identity function $s(x) = x$ on V.

It then follows that for any term t, $\bar{s}(t) = t$. For any wff φ, let φ^* be the result of replacing the equality symbol in φ by E. Then

$$\models_{\mathfrak{A}} \varphi^*[s] \quad \text{iff} \quad \varphi \in \Delta.$$

Details: That $\bar{s}(t) = t$ can be proved by induction on t, but the proof is entirely straightforward.

The other claim, that

$$\models_{\mathfrak{A}} \varphi^*[s] \quad \text{iff} \quad \varphi \in \Delta,$$

we prove by induction on the number of places at which connective or quantifier symbols appear in φ.

Case 1: Atomic formulas. We defined $\mathfrak{A}$ in such a way as to make this case immediate. For example, if φ is Pt, then

$$\begin{aligned} \models_{\mathfrak{A}} Pt[s] \quad &\text{iff} \quad \bar{s}(t) \in P^{\mathfrak{A}}, \\ &\text{iff} \quad t \in P^{\mathfrak{A}}, \\ &\text{iff} \quad Pt \in \Delta. \end{aligned}$$

Similarly,

$$\begin{aligned} \models_{\mathfrak{A}} uEt[s] \quad &\text{iff} \quad \langle \bar{s}(u), \bar{s}(t) \rangle \in E^{\mathfrak{A}}, \\ &\text{iff} \quad \langle u, t \rangle \in E^{\mathfrak{A}}, \\ &\text{iff} \quad u = t \in \Delta. \end{aligned}$$

Case 2: Negation.

$$\begin{aligned} \models_{\mathfrak{A}} (\neg \varphi)^*[s] \quad &\text{if} \quad \not\models_{\mathfrak{A}} \varphi^*[s], \\ &\text{iff} \quad \varphi \notin \Delta \quad \text{by inductive hypothesis,} \\ &\text{iff} \quad (\neg \varphi) \in \Delta \quad \text{by properties of } \Delta. \end{aligned}$$

Case 3: Conditional.

$$\begin{aligned} \models_{\mathfrak{A}} (\varphi \to \psi)^*[s] \quad &\text{iff} \quad \not\models_{\mathfrak{A}} \varphi^*[s] \text{ or } \models_{\mathfrak{A}} \psi^*[s], \\ &\text{iff} \quad \varphi \notin \Delta \text{ or } \psi \in \Delta \text{ by inductive hypothesis,} \\ &\text{iff} \quad (\neg \varphi) \in \Delta \text{ or } \psi \in \Delta, \\ &\Rightarrow \Delta \vdash (\varphi \to \psi), \text{ in fact tautologically,} \\ &\Rightarrow \varphi \notin \Delta \text{ or } [\varphi \in \Delta \text{ and } \Delta \vdash \psi], \\ &\Rightarrow (\neg \varphi) \in \Delta \text{ or } \psi \in \Delta, \end{aligned}$$

which closes the loop. And

$$\Delta \vdash (\varphi \to \psi) \quad \text{iff} \quad (\varphi \to \psi) \in \Delta.$$

(This should be compared with Exercise 2 of Section 1.7.)

Case 4: Quantification. We want to show that

$$\models_{\mathfrak{A}} \forall x \, \varphi^*[s] \quad \text{iff} \quad \forall x \, \varphi \in \Delta.$$

(The notational ambiguity is harmless since $\forall x(\varphi^*)$ is the same as $(\forall x \, \varphi)^*$.) Δ includes the wff θ:

$$\neg \forall x \, \varphi \rightarrow \neg \varphi_c^x.$$

To show that

$$\models_{\mathfrak{A}} \forall x \, \varphi^*[s] \Rightarrow \forall x \, \varphi \in \Delta,$$

we can argue: If φ^* is true of everything, then it is true of c, whence by the inductive hypothesis $\varphi_c^x \in \Delta$. But then $\forall x \, \varphi \in \Delta$, because c was chosen to be a counterexample to φ if there was one. In more detail:

$$\models_{\mathfrak{A}} \forall x \, \varphi^*[s] \Rightarrow \models_{\mathfrak{A}} \varphi^*[s(x \mid c)]$$

$$\begin{aligned}
&\Rightarrow \models_{\mathfrak{A}} (\varphi^*)_c^x[s] && \text{by the substitution lemma} \\
&\Rightarrow \models_{\mathfrak{A}} (\varphi_c^x)^*[s], && \text{this being the same formula} \\
&\Rightarrow \varphi_c^x \in \Delta && \text{by the inductive hypothesis} \\
&\Rightarrow (\neg \varphi_c^x) \notin \Delta && \text{by consistency} \\
&\Rightarrow (\neg \forall x \, \varphi) \notin \Delta && \text{since } \theta \in \Delta \text{ and } \Delta \text{ is} \\
& && \quad \text{deductively closed}
\end{aligned}$$

$$\Rightarrow \forall x \, \varphi \in \Delta.$$

(This is our only use of Θ. We needed to know that if $(\neg \forall x \, \varphi) \in \Delta$, then for a particular c we would have $(\neg \varphi_c^x) \in \Delta$.)

We turn now to the converse. We can almost argue as follows:

$$\begin{aligned}
\not\models_{\mathfrak{A}} \forall x \, \varphi^*[s] &\Rightarrow \not\models_{\mathfrak{A}} \varphi^*[s(x \mid t)] && \text{for some } t \\
&\rightsquigarrow \not\models_{\mathfrak{A}} (\varphi_t^x)^*[s] && \text{by the substitution lemma} \\
&\Rightarrow \varphi_t^x \notin \Delta && \text{by the inductive hypothesis} \\
&\rightsquigarrow \forall x \, \varphi \notin \Delta && \text{since } \Delta \text{ is deductively closed.}
\end{aligned}$$

The flaw here is that the two wavy implications require that t be substitutable for x in φ. This may not be the case, but we can use the usual repair: We change to an alphabetic variant ψ of φ in which t is substitutable for x. Then

$$\begin{aligned}
\not\models_{\mathfrak{A}} \forall x \, \varphi^*[s] &\Rightarrow \not\models_{\mathfrak{A}} \varphi^*[s(x \mid t)] && \text{for some } t, \text{ henceforth fixed} \\
&\Rightarrow \not\models_{\mathfrak{A}} \psi^*[s(x \mid t)] && \text{by the semantical equivalence of} \\
& && \quad \text{alphabetic variants (Corollary 25D)} \\
&\Rightarrow \not\models_{\mathfrak{A}} (\psi_t^x)^*[s] && \text{by the substitution lemma} \\
&\Rightarrow \psi_t^x \notin \Delta && \text{by the inductive hypothesis} \\
&\Rightarrow \forall x \, \psi \notin \Delta && \text{since } \Delta \text{ is deductively closed} \\
&\Rightarrow \forall x \, \varphi \notin \Delta && \text{by the syntactical equivalence of} \\
& && \quad \text{alphabetic variants (Theorem 24I).}
\end{aligned}$$

This completes the list of possible cases; it now follows by induction that for any φ,

$$\models_{\mathfrak{A}} \varphi^*[s] \quad \text{iff} \quad \varphi \in \Delta.$$

If our original language did not include the equality symbol, then we are done. For we need only restrict $\mathfrak{A}$ to the original language to obtain a structure that satisfies every member of Γ with the identity function.

But now assume that the equality symbol is in the language. Then $\mathfrak{A}$ will no longer serve. For example, if Γ contains the sentence $c = d$ (where c and d are distinct constant symbols), then we need a structure $\mathfrak{B}$ in which $c^{\mathfrak{B}} = d^{\mathfrak{B}}$. We obtain $\mathfrak{B}$ as the quotient structure $\mathfrak{A}/E$ of $\mathfrak{A}$ modulo $E^{\mathfrak{A}}$.

STEP 5: $E^{\mathfrak{A}}$ is an equivalence relation on $|\mathfrak{A}|$. For each t in $|\mathfrak{A}|$ let $[t]$ be its equivalence class. $E^{\mathfrak{A}}$ is, in fact, a *congruence relation* for $\mathfrak{A}$. This means that the following conditions are met:

(i) $E^{\mathfrak{A}}$ is an equivalence relation on $|\mathfrak{A}|$.
(ii) $P^{\mathfrak{A}}$ is compatible with $E^{\mathfrak{A}}$ for each predicate symbol P:

$$\langle t_1, \ldots, t_n \rangle \in P^{\mathfrak{A}} \text{ and } t_i E^{\mathfrak{A}} t_i' \text{ for } 1 \leq i \leq n \Rightarrow \langle t_1', \ldots, t_n' \rangle \in P^{\mathfrak{A}}.$$

(iii) $f^{\mathfrak{A}}$ is compatible with $E^{\mathfrak{A}}$ for each function symbol f:

$$t_i E^{\mathfrak{A}} t_i' \text{ for } 1 \leq i \leq n \Rightarrow f^{\mathfrak{A}}(t_1, \ldots, t_n) E^{\mathfrak{A}} f^{\mathfrak{A}}(t_1', \ldots, t_n').$$

Under these circumstances we can form the quotient structure $\mathfrak{A}/E$, defined as follows:

(a) $|\mathfrak{A}/E|$ is the set of all equivalence classes of members of $|\mathfrak{A}|$.
(b) For each n-place predicate symbol P,

$$\langle [t_1], \ldots, [t_n] \rangle \in P^{\mathfrak{A}/E} \text{ iff } \langle t_1, \ldots, t_n \rangle \in P^{\mathfrak{A}}.$$

(c) For each n-place function symbol f,

$$f^{\mathfrak{A}/E}([t_1], \ldots, [t_n]) = [f^{\mathfrak{A}}(t_1, \ldots, t_n)].$$

This includes the $n = 0$ cases:

$$c^{\mathfrak{A}/E} = [c^{\mathfrak{A}}].$$

Let $h : |\mathfrak{A}| \rightarrow |\mathfrak{A}/E|$ be the natural map:

$$h(t) = [t].$$

Then h is a homomorphism of $\mathfrak{A}$ onto $\mathfrak{A}/E$. Furthermore, $E^{\mathfrak{A}/E}$ is the equality relation on $|\mathfrak{A}/E|$. Consequently, for any φ:

$$\varphi \in \Delta \Leftrightarrow \models_{\mathfrak{A}} \varphi^*[s]$$
$$\Leftrightarrow \models_{\mathfrak{A}/E} \varphi^*[h \circ s]$$
$$\Leftrightarrow \models_{\mathfrak{A}/E} \varphi[h \circ s]$$

So $\mathfrak{A}/E$ satisfies every member of Δ (and hence every member of Γ) with $h \circ s$.

Details: Recall that

$$t E^{\mathfrak{A}} t' \quad \text{iff} \quad (t = t') \in \Delta,$$
$$\text{iff} \quad \Delta \vdash t = t'.$$

(i) $E^{\mathfrak{A}}$ is an equivalence relation on $\mathfrak{A}$ by properties Eq1, Eq2, and Eq3 of equality.

(ii) $P^{\mathfrak{A}}$ is compatible with $E^{\mathfrak{A}}$ by property Eq4 of equality.

(iii) $f^{\mathfrak{A}}$ is compatible with $E^{\mathfrak{A}}$ by property Eq5 of equality.

It then follows from the compatibility of $P^{\mathfrak{A}}$ with $E^{\mathfrak{A}}$ that $P^{\mathfrak{A}/E}$ is well defined. Similarly, $f^{\mathfrak{A}/E}$ is well defined because $f^{\mathfrak{A}}$ is compatible with $E^{\mathfrak{A}}$.

It is immediate from the construction that h is a homomorphism of $\mathfrak{A}$ onto $\mathfrak{A}/E$. And

$$[t] E^{\mathfrak{A}/E} [t'] \quad \text{iff} \quad t E^{\mathfrak{A}} t',$$
$$\text{iff} \quad [t] = [t'].$$

Finally,

$$\varphi \in \Delta \Leftrightarrow \models_{\mathfrak{A}} \varphi^*[s] \qquad \text{by step 4}$$
$$\Leftrightarrow \models_{\mathfrak{A}/E} \varphi^*[h \circ s] \quad \text{by the homomorphism theorem}$$
$$\Leftrightarrow \models_{\mathfrak{A}/E} \varphi[h \circ s],$$

the last step being justified by the fact that $E^{\mathfrak{A}/E}$ is the equality relation on $|\mathfrak{A}/E|$.

> STEP 6:   Restrict the structure $\mathfrak{A}/E$ to the original language. This restriction of $\mathfrak{A}/E$ satisfies every member of Γ with $h \circ s$. $\dashv$

For an uncountable language, a few modifications to the foregoing proof of the completeness theorem are needed. Say that the language has cardinality λ. (By this we mean that it has λ symbols or, equivalently, λ formulas.) We will describe the modifications needed, assuming the reader has a substantial knowledge of set theory. In step 1 we add λ new constant symbols; the details remain unchanged. In step 2, only the details change. The cardinal λ is an initial ordinal. (We have tacitly well ordered the language here.) "Enumerate" the pairs

$$\langle \varphi_\alpha, x_\alpha \rangle_{\alpha < \lambda}$$

indexed by ordinals less than λ. For $\alpha < \lambda$, θ_α is

$$\neg \forall x_\alpha \varphi_\alpha \rightarrow (\neg \varphi)^{x_\alpha}_{c_\alpha},$$

where c_α is the first of the new constant symbols not in φ_α or in θ_β for any $\beta < \alpha$. (This excludes at most $\aleph_0 \cdot \mathrm{card}(\alpha)$ constant symbols, so there are some left.) Finally, in step 3, we can obtain the maximal set Δ by use of Zorn's lemma. The rest of the proof remains unchanged.

COMPACTNESS THEOREM   (a) If $\Gamma \models \varphi$, then for some finite $\Gamma_0 \subseteq \Gamma$ we have $\Gamma_0 \models \varphi$.

(b) If every finite subset Γ_0 of Γ is satisfiable, then Γ is satisfiable.

In particular, a set Σ of sentences has a model iff every finite subset has a model.

PROOF   To prove part (a) of the compactness theorem, we simply observe that

$\Gamma \models \varphi \Rightarrow \Gamma \vdash \varphi$
$\qquad \Rightarrow \Gamma_0 \vdash \varphi$ for some finite $\Gamma_0 \subseteq \Gamma$, deductions being finite
$\qquad \Rightarrow \Gamma_0 \models \varphi$.

Part (b) has a similar proof. If every finite subset of Γ is satisfiable, then by soundness every finite subset of Γ is consistent. Thus Γ is consistent, since deductions are finite. So by completeness, Γ is satisfiable. (Actually parts (a) and (b) are equivalent; cf. Exercise 3 of Section 1.7.)   ⊣

When a person first hears of the compactness theorem, his natural inclination is to try to combine (by some algebraic or set-theoretic operation) the structures in which the various finite subsets are satisfied, in such a way as to obtain a structure in which the entire set is satisfied. In fact, such a proof *is* possible; the operation to use is the *ultraproduct* construction. But we will refrain from digressing further into this intriguing possibility.

Notice that the compactness theorem involves only semantical notions of Section 2.2; it does not involve deductions at all. And there are proofs that avoid deductions. The same remarks apply to the following theorem.

*ENUMERABILITY THEOREM   For a reasonable language, the set of valid wffs can be effectively enumerated.

By a reasonable language we mean one whose set of parameters can be effectively enumerated and such that the two relations

$$\{\langle P, n \rangle \mid P \text{ is an } n\text{-place predicate symbol}\}$$

and

$$\{\langle f, n \rangle \mid f \text{ is an } n\text{-place function symbol}\}$$

are decidable. For example, any language with only finitely many parameters (such a language will be called a *finite language*) is certainly reasonable, because finite sets are always decidable. On the other hand, a reasonable language must be countable, since we cannot effectively enumerate an uncountable set. (In fact, a stronger statement applies: As

in Section 1.7, a suitable input/output format is needed in which the underlying set of symbols communicated is *finite*, which implies that the set of all strings is countable.)

A precise version of this theorem will be given in Section 3.4. (See especially item 20 there.) The proofs of the two versions are in essence the same.

PROOF.   The essential fact is that Λ, and hence the set of deductions, is decidable.

Suppose that we are given some expression ε. (The assumption of reasonableness enters already here. There are only countably many things eligible to be given by one person to another.) We want to decide whether or not ε is in Λ. First we check that ε has the syntactical form necessary to be a formula. (For sentential logic we gave detailed instructions for such a check; see Section 1.3. Similar instructions can be given for first-order languages, by using Section 2.3.) If ε passes that test, we then check (by constructing a truth table) to see if ε is a generalization of a tautology. If not, then we proceed to see if ε has the syntactical form necessary to be in axiom group 2. And so forth. If ε has not been accepted by the time we finish with axiom group 6, then ε is not in Λ.

(The above is intended to convince the reader that he really can tell members of Λ from nonmembers. The reader who remains dubious can look forward to the rerun in Section 3.4.)

Since Λ is decidable, the set of tautological consequences of Λ is effectively enumerable; see Theorem 17G. But

$$\{\alpha \mid \alpha \text{ is a tautological consequence of } \Lambda\}$$
$$= \{\alpha \mid \vdash \alpha\} \text{ by Theorem 24B,}$$
$$= \{\alpha \mid \alpha \text{ is valid}\}. \qquad \dashv$$

An alternative to the last paragraph of this proof is the following argument, which is possibly more illuminating: First we claim that the set of deductions (from $\varnothing$) is decidable. For given a finite sequence $\alpha_0, \ldots, \alpha_n$ we can examine each α_i in turn to see if it is in Λ or is obtainable by modus ponens from earlier members of the sequence. Then to enumerate the validities, we begin by enumerating all finite sequences of wffs. We look at each sequence as it is produced and decide whether or not it is a deduction. If not, we discard it. But if it is, then we put its last member on the list of validities. Continuing in this way, we generate — in an inefficient way — a list on which any valid formula will eventually appear.

*COROLLARY 25F   Let Γ be a decidable set of formulas in a reasonable language.

    (a) The set of theorems of Γ is effectively enumerable.

    (b) The set $\{\varphi \mid \Gamma \models \varphi\}$ of formulas logically implied by Γ is effectively enumerable.

(Of course parts (a) and (b) refer to the same set. This corollary includes the enumerability theorem itself, in which $\Gamma = \varnothing$.)

PROOF 1.   Enumerate the validities; whenever you find one of the form

$$\alpha_n \longrightarrow \cdots \longrightarrow \alpha_1 \longrightarrow \alpha_0,$$

check to see if $\alpha_n, \ldots, \alpha_1$ are in Γ. If so, then put α_0 on the list of theorems of Γ. In this way, any theorem of Γ is eventually listed.

⊣

PROOF 2. $\Gamma \cup \Lambda$ is decidable, so its set of tautological consequences is effectively enumerable. And that is just the set we want.   ⊣

For example, let Γ be the (decidable) set of axioms for any of the usual systems of set theory. Then this corollary tells us that the set of theorems of set theory is effectively enumerable.

More generally, in setting up some axiomatic theory, it is natural to insist that the set of axioms be decidable. After all, we want proofs from these axioms to be *convincing* arguments that can be *verified*. Part of the verification process involves checking that statements alleged to be axioms are indeed axioms. For this to be possible, the set of axioms needs to be decidable (or at least semidecidable). This has the consequence that the set of theorems that follow from the axioms is effectively enumerable.

⋆**COROLLARY 25G**   Assume that Γ is a decidable set of formulas in a reasonable language, and for any sentence σ either $\Gamma \models \sigma$ or $\Gamma \models \neg \sigma$. Then the set of sentences implied by Γ is decidable.

PROOF.   If Γ is inconsistent, then we have simply the (decidable) set of all sentences. So assume that Γ is consistent. Suppose that we are given a sentence σ and asked to decide whether or not $\Gamma \models \sigma$. We can enumerate the theorems of Γ and look for σ or $\neg \sigma$. Eventually one will appear, and then we know the answer.   ⊣

(Observe that this proof actually describes two decision procedures. One is correct when Γ is inconsistent, the other is correct when Γ is consistent. So in either case a decision procedure exists. But we cannot necessarily determine effectively, given a finite description of Γ, which one is to be used. A set is decidable if there *exists* a decision procedure for it. That is not the same as having a known decision procedure in our hands.)

It should be remarked that our proofs of enumerability cannot, in general, be strengthened to proofs of decidability. For almost all languages the set of validities is *not* decidable. (See Church's Theorem, Section 3.5.)

Historical Notes

The completeness theorem (for countable languages) was contained in the 1930 doctoral dissertation of Kurt Gödel. (It is not to be confused with the "Gödel incompleteness theorem," published in 1931. We will consider this latter result in Chapter 3.) The compactness theorem (for countable languages) was given as a corollary.

The compactness theorem for uncountable languages was implicit in a 1936 paper by Anatolii Mal′cev. His proof used Skolem functions (cf. Section 4.2) and the compactness theorem of sentential logic. The first explicit statement of the compactness theorem for uncountable languages was in a 1941 paper by Mal′cev.

The enumerability theorem, as well as following from Gödel's 1930 work, was also implicit in results published in 1928 by Thoralf Skolem.

The proof we have given for the completeness theorem is patterned after one given by Leon Henkin in his dissertation, published in 1949. Unlike Gödel's original proof, Henkin's proof generalizes easily to languages of any cardinality.

Exercises

1. (Semantical rule EI) Assume that the constant symbol c does not occur in φ, ψ, or Γ, and that $\Gamma; \varphi_c^x \models \psi$. Show (without using the soundness and completeness theorems) that $\Gamma; \exists x \, \varphi \models \psi$.

2. Prove the equivalence of parts (a) and (b) of the completeness theorem. *Suggestion*: $\Gamma \models \varphi$ iff $\Gamma \cup \{\neg \varphi\}$ is unsatisfiable. And Δ is satisfiable iff $\Delta \not\models \bot$, where $\bot$ is some unsatisfiable, refutable formula like $\neg \forall x \, x = x$.

    Remark: Similarly, the soundness theorem is equivalent to the statement that every satisfiable set of formulas is consistent.

3. Assume that $\Gamma \vdash \varphi$ and that P is a predicate symbol which occurs neither in Γ nor in φ. Is there a deduction of φ from Γ in which P nowhere occurs? *Suggestion*: There are two very different approaches to this problem. The "soft" approach makes use of two languages, one that contains P and one that does not. The "hard" approach considers the question whether P can be systematically eliminated from a given deduction of φ from Γ.

4. Let $\Gamma = \{\neg \forall v_1 P v_1, P v_2, P v_3, \ldots\}$. Is Γ consistent? Is Γ satisfiable?

5. Show that an infinite map can be colored with four colors iff every finite submap of it can be. *Suggestion*: Take a language having a constant symbol for each country and having four one-place predicate symbols for the colors. Use the compactness theorem.

6. Let Σ_1 and Σ_2 be sets of sentences such that nothing is a model of both Σ_1 and Σ_2. Show that there is a sentence τ such that Mod $\Sigma_1 \subseteq$ Mod τ and Mod $\Sigma_2 \subseteq$ Mod $\neg \tau$. (This can be stated: Disjoint EC_Δ classes can be separated by an EC class.) *Suggestion*: $\Sigma_1 \cup \Sigma_2$ is unsatisfiable; apply compactness.

7. The completeness theorem tells us that each sentence either has a deduction (from $\varnothing$) or has a counter-model (i.e., a structure in which it is false). For each of the following sentences, either show there is a deduction or give a counter-model.
    (a) $\forall x(Qx \to \forall y\, Qy)$
    (b) $(\exists x\, Px \to \forall y\, Qy) \to \forall z(Pz \to Qz)$
    (c) $\forall z(Pz \to Qz) \to (\exists x\, Px \to \forall y\, Qy)$
    (d) $\neg \exists y \forall x(Pxy \leftrightarrow \neg Pxx)$

8. Assume the language (with equality) has just the parameters $\forall$ and P, where P is a two-place predicate symbol. Let $\mathfrak{A}$ be the structure with $|\mathfrak{A}| = \mathbb{Z}$, the set of integers (positive, negative, and zero) and with $\langle a, b\rangle \in P^{\mathfrak{A}}$ iff $|a - b| = 1$. Thus $\mathfrak{A}$ looks like an infinite graph:

$$\cdots \longleftrightarrow \bullet \longleftrightarrow \bullet \longleftrightarrow \bullet \longleftrightarrow \cdots$$

Show that there is an elementarily equivalent structure $\mathfrak{B}$ that is not connected. (Being *connected* means that for every two members of $|\mathfrak{B}|$, there is a path between them. A *path* — of length n — from a to b is a sequence $\langle p_0, p_1, \ldots, p_n\rangle$ with $a = p_0$ and $b = p_n$ and $\langle p_i, p_{i+1}\rangle \in P^{\mathfrak{B}}$ for each i.) *Suggestion*: Add constant symbols c and d. Write down sentences saying c and d are far apart. Apply compactness.

9. In Section 2.4 we used a certain set Λ of logical axioms. That set can be altered, within limits.
    (a) Suppose we add to Λ some formula ψ that is not valid. Show that the soundness theorem now fails.
    (b) At the other extreme, suppose we take no logical axioms at all: $\Lambda = \varnothing$. Show that the completeness theorem now fails.
    (c) Suppose we modify Λ by adding one new valid formula. Explain why both the soundness theorem and the completeness theorem still hold.

SECTION 2.6
Models of Theories

In this section we will leave behind deductions and logical axioms. Instead we return to topics discussed in Section 2.2. But now, in the presence of the theorems of the preceding section, we will be able to answer more questions than we could before.

Finite Models

Some sentences have only infinite models, for example, the sentence saying that $<$ is an ordering with no largest element. The negation of such a sentence is *finitely valid*, that is, it is true in every finite structure.

It is also possible to have sentences having only finite models. For example, any model of $\forall x \, \forall y \, x = y$ has cardinality 1. But if all models of Σ are finite, then there is a finite bound on the size of the models, by the following theorem.

> **THEOREM 26A**   If a set Σ of sentences has arbitrarily large finite models, then it has an infinite model.

> PROOF.   For each integer $k \geq 2$, we can find a sentence λ_k that translates, "There are at least k things." For example,
>
> $$\lambda_2 = \exists v_1 \, \exists v_2 \, v_1 \neq v_2,$$
> $$\lambda_3 = \exists v_1 \, \exists v_2 \, \exists v_3 (v_1 \neq v_2 \wedge v_1 \neq v_3 \wedge v_2 \neq v_3).$$
>
> Consider the set
>
> $$\Sigma \cup \{\lambda_2, \lambda_3, \ldots\}.$$
>
> By hypothesis any finite subset has a model. So by compactness the entire set has a model, which clearly must be infinite.   ⊣

For example, it is a priori conceivable that there might be some very subtle equation of group theory that was true in every finite group but false in every infinite group. But by the above theorem, no such equation exists.

The proof to this theorem illustrates a useful method for obtaining a structure with given properties. One writes down sentences (possibly in an expanded language) stating the properties one wants. One then argues that any finite subset of the sentences has a model. The compactness theorem does the rest. We will see more examples of this method in the coming pages.

> **COROLLARY 26B**   The class of all finite structures (for a fixed language) is not EC_Δ. The class of all infinite structures is not EC.

PROOF.   The first sentence follows immediately from the theorem. If the class of all infinite structures is Mod τ, then the class of all finite structures is Mod $\neg\, \tau$. But this class isn't even EC_Δ, much less EC.   ⊣

The class of infinite structures *is* EC_Δ, being Mod$\{\lambda_2, \lambda_3, \ldots\}$.

Next we want to consider decision problems connected with finite structures. For any structure $\mathfrak{A}$, define the *theory* of $\mathfrak{A}$, written Th $\mathfrak{A}$, to be the set of all sentences true in $\mathfrak{A}$. For a *finite* structure $\mathfrak{A}$, is Th $\mathfrak{A}$ decidable? Is the set of sentences having finite models decidable?

The following observations will help here:

1. Any finite structure $\mathfrak{A}$ is isomorphic to a structure with universe $\{1, 2, \ldots, n\}$ where n is the size of $\mathfrak{A}$ (i.e., $n = \operatorname{card} |\mathfrak{A}|$). The idea here is, where $|\mathfrak{A}| = \{a_1, \ldots, a_n\}$, simply to replace a_i by i.

For example, suppose the language has only the parameters $\forall$ and a two-place predicate symbol E (for the "edge" relation in a directed graph). Consider the finite structure $\mathfrak{B}$ with universe $|\mathfrak{B}|$ consisting of a set of four distinct objects $\{a, b, c, d\}$, and with

$$E^{\mathfrak{B}} = \{\langle a, b \rangle, \langle b, a \rangle, \langle b, c \rangle, \langle c, c \rangle\}.$$

Then $\mathfrak{B}$ is isomorphic to the structure

$$(\{1, 2, 3, 4\}; \{\langle 1, 2 \rangle, \langle 2, 1 \rangle, \langle 2, 3 \rangle, \langle 3, 3 \rangle\}).$$

But there are other possibilities here; if we had focused on the members of $|\mathfrak{B}|$ in the order b, a, d, c then we would have arrived at the isomorphic (but different) structure

$$(\{1, 2, 3, 4\}; \{\langle 1, 2 \rangle, \langle 2, 1 \rangle, \langle 1, 4 \rangle, \langle 4, 4 \rangle\}).$$

2. A finite structure of the sort just described can, for a finite language, be specified by a finite string of symbols. In the example,

$$(\{1, 2, 3, 4\}; \{\langle 1, 2 \rangle, \langle 2, 1 \rangle, \langle 2, 3 \rangle, \langle 3, 3 \rangle\}),$$

the above line completely specifies the structure, and it can be written down with numerals in base 10 (or your favorite base) along with punctuation and delimiters (e.g., parentheses). Therefore such a structure can be *communicated* to another person or to a machine. The finite string of symbols can be written down in a suitable input format.

3. Given a finite structure $\mathfrak{A}$ for a finite language, with universe $\{1, \ldots, n\}$ (and by the preceding observation, it *is* possible to be given such an object), a wff φ, and an assignment s_φ of numbers in this universe to the variables free in φ (there are only finitely many, of course), we can effectively decide whether or not $\models_{\mathfrak{A}} \varphi[s_\varphi]$.

For example, given

$$\mathfrak{B} = (\{1, 2, 3, 4\}; \{\langle 1, 2 \rangle, \langle 2, 1 \rangle, \langle 2, 3 \rangle, \langle 3, 3 \rangle\})$$
$$\varphi = \forall\, v_1((\neg\, \forall\, v_2 \neg E v_2 v_1) \rightarrow E v_1 v_1),$$

we can organize the computation in the tree form shown in Fig. 9. We find that the sentence φ, which says "Anything in the range of E is related to itself," is false in $\mathfrak{B}$.
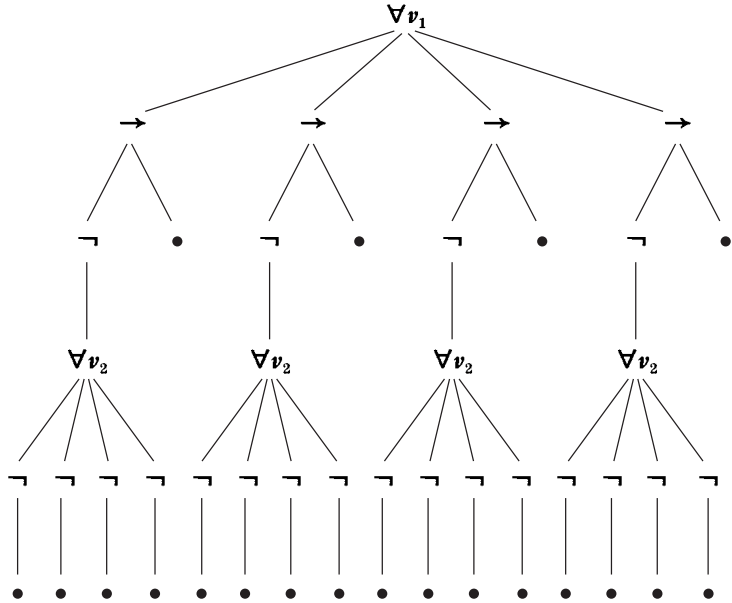
Figure 9. Checking the sentence $\forall v_1((\neg\forall v_2\neg E v_2 v_1) \rightarrow E v_1 v_1)$ in a structure with universe of size 4.

At each leaf in the tree (i.e., each minimal vertex), we have an atomic formula and we do a "table look-up" to see if it is satisfied. Notice that each quantifier triggers a search through the n-element universe. For a formula φ with k quantifiers, the number of leaves in the tree will be bounded by a polynomial in n of degree k. If the language contains function symbols, then each term needs to be evaluated, using the (finite) function provided by the structure.

In particular, restricting ourselves to sentences, we can effectively decide, given $\mathfrak{A}$ as above and a sentence σ, whether or not $\mathfrak{A}$ is a model of σ. (In fact here σ could even be a second-order sentence, as in Chapter 4.)

    ★THEOREM 26C   For a finite structure $\mathfrak{A}$ in a finite language, Th $\mathfrak{A}$ is decidable.

PROOF 1.   By observation 1, we can replace $\mathfrak{A}$ by an isomorphic structure with universe of the form $\{1, \ldots, n\}$, without changing which sentences are true. Then apply observation 3.   ⊣

PROOF 2.   By Exercise 17(a) of Section 2.2, there is a sentence $\delta_\mathfrak{A}$ that specifies $\mathfrak{A}$ up to isomorphism. It follows that

$$\mathrm{Th}\,\mathfrak{A} = \{\sigma \mid \delta_\mathfrak{A} \models \sigma\}.$$

(Details: For "$\subseteq$" note that if σ is true in $\mathfrak{A}$, then it is true in all isomorphic copies, and hence in all models of $\delta_\mathfrak{A}$. So $\delta_\mathfrak{A} \models \sigma$. The other direction is simpler; if $\delta_\mathfrak{A} \models \sigma$ then σ is true in all models of $\delta_\mathfrak{A}$, of which $\mathfrak{A}$ is one.) Apply Corollary 25G, noting that for each σ, either $\models_\mathfrak{A} \sigma$ or $\models_\mathfrak{A} \neg\sigma$.   ⊣

4. Given a sentence σ and a positive integer n, we can effectively decide whether or not σ has an n-element model. That is, the binary relation

$$\{\langle \sigma, n \rangle \mid \sigma \text{ has a model of size } n\}$$

is decidable.

The key idea is that there are only finitely many structures to check, and we can do that. The sentence σ has a model of size n if and only if it has a model with universe $\{1, \ldots, n\}$ by observation 1. With the language restricted to the parameters that occur in σ, there are only finitely many such structures, and we can systematically generate all of them. (For example, if the only parameters are $\forall$ and a 2-place predicate symbol, then there are 2^{n^2} different structures.) Applying observation 3, we test to see if any of these are models of σ.

5. The *spectrum* of a sentence σ is defined to be $\{n \mid \sigma$ has a model of size $n\}$. See Exercise 16 in Section 2.2. It follows from observation 4 that the spectrum of any sentence is a decidable set of positive integers.

⋆THEOREM 26D   For a finite language, $\{\sigma \mid \sigma$ has a finite model$\}$ is effectively enumerable.

PROOF.   Here is a semidecision procedure: Given σ, first check to see if it has a model of size 1, by using observation 4. If not, try size 2. Keep going.   ⊣

⋆COROLLARY 26E   Assume the language is finite, and let Φ be the set of sentences true in every *finite* structure. Then its complement, $\overline{\Phi}$, is effectively enumerable.

PROOF.   For a sentence σ,

$$\sigma \in \overline{\Phi} \iff (\neg\sigma) \text{ has a finite model.}$$

We can apply the above semidecision procedure to $(\neg\sigma)$.   ⊣

It follows (by Theorem 17F) that Φ is decidable if and only if it is effectively enumerable. But this does not happen. We state without proof the following:

*TRAKHTENBROT'S THEOREM (1950)   The set of sentences

$$\Phi = \{\sigma \mid \sigma \text{ is true in every finite structure}\}$$

is not in general decidable or effectively enumerable.

Thus the analogue of the enumerability theorem for finite structures only is false.

Size of Models

In the proof of the completeness theorem in Section 2.5, we started with a consistent set Γ and formed a structure $\mathfrak{A}/E$ in which it was satisfied. How large was that structure? We claim that if our initial language was countable, then $|\mathfrak{A}/E|$ is a countable set. Thus a consistent set of sentences in a countable language has a countable model.

$\mathfrak{A}/E$ was constructed from a preliminary structure $\mathfrak{A}$. The universe of $\mathfrak{A}$ was the set of all terms in the language obtained by adding a countable set of new constant symbols. But the augmented language was still countable, so the set of all expressions (and hence the set of all terms) was countable. That is, $|\mathfrak{A}|$ was countable.

The universe of $\mathfrak{A}/E$ consisted of equivalence classes of members of $\mathfrak{A}$, so it, too, is a countable set. (We can map $|\mathfrak{A}/E|$ one-to-one into $|\mathfrak{A}|$ by assigning to each equivalence class some chosen member.) The conclusion is that $\mathfrak{A}/E$ is a countable structure, as claimed.

LÖWENHEIM–SKOLEM THEOREM (1915)   (a) Let Γ be a satisfiable set of formulas in a countable language. Then Γ is satisfiable in some countable structure.

(b) Let Σ be a set of sentences in a countable language. If Σ has any model, then it has a countable model.

PROOF.   First observe that Γ is consistent, by the soundness theorem. Then by the completeness theorem (plus the foregoing remarks) it can be satisfied in a countable structure.   ⊣

(There is another, more direct proof of this theorem that will be indicated in Section 4.2; see especially Exercise 1 there. That proof, which does not use a deductive calculus, begins with an arbitrary structure $\mathfrak{A}$ in which Γ can be satisfied, and by various manipulations extracts from it a suitable countable *substructure* in which Γ is still satisfied.)

The Löwenheim–Skolem theorem was published by Leopold Löwenheim in 1915 for the case where Γ is a singleton; Thoralf Skolem in 1920 extended this to a possibly infinite Γ. The theorem marked a new phase in mathematical logic. Earlier work had been done in the direction

of *formalizing* mathematics by means of formal languages and deductive calculi; this work was initiated largely by Gottlob Frege in 1879. For example, the *Principia Mathematica* (1910–1913) of Whitehead and Russell carried out such a formalization in great detail. But the modern phase began when logicians stepped back and began to prove results *about* the formal systems they had been constructing. Other early work in this trend was done by David Hilbert, Emil Post, Kurt Gödel (as mentioned before), Alfred Tarski, and others.

For a sample application of the Löwenheim–Skolem theorem, let A_{ST} be your favorite set of axioms for set theory. We certainly hope these axioms are consistent. And so they have some model. By the Löwenheim–Skolem theorem, the axioms have a countable model $\mathfrak{S}$. Of course, $\mathfrak{S}$ is also a model of all the sentences logically implied by A_{ST}. One of these sentences asserts (when translated back into English according to the intended translation) that there are uncountably many sets. There is no contradiction here, but the situation is sufficiently puzzling to be called "Skolem's paradox." It is true that *in the structure* $\mathfrak{S}$ there is no point that satisfies the formal definition of being a one-to-one map of the natural numbers onto the universe. But this in no way excludes the possibility of there being (outside $\mathfrak{S}$) some genuine function providing such a one-to-one correspondence.

Recall that the *theory* of $\mathfrak{A}$, written Th $\mathfrak{A}$, is the set of all sentences true in $\mathfrak{A}$. We can apply the Löwenheim–Skolem theorem (with $\Sigma =$ Th $\mathfrak{A}$) to prove that for any structure $\mathfrak{A}$ for a countable language, there is a *countable* elementarily equivalent structure $\mathfrak{B}$. If $\mathfrak{B}$ is a model of Th $\mathfrak{A}$ then $\mathfrak{A} \equiv \mathfrak{B}$ because

$$\models_{\mathfrak{A}} \sigma \;\Rightarrow\; \sigma \in \text{Th}\,\mathfrak{A} \;\Rightarrow\; \models_{\mathfrak{B}} \sigma$$

and

$$\not\models_{\mathfrak{A}} \sigma \Rightarrow \models_{\mathfrak{A}} \neg\sigma \Rightarrow (\neg\sigma) \in \text{Th}\,\mathfrak{A} \Rightarrow \models_{\mathfrak{B}} \neg\sigma \Rightarrow \not\models_{\mathfrak{B}} \sigma.$$

For example, the real field $(\mathbb{R}; 0, 1, +, \cdot)$ is an uncountable structure for a finite language. Therefore there must be some countable structure (also a field) satisfying exactly the same sentences. (In fact Tarski showed we can take the field of algebraic real numbers for this.)

EXAMPLE.   Consider the structure

$$\mathfrak{N} = (\mathbb{N}; 0, S, <, +, \cdot).$$

We claim that there is a countable structure $\mathfrak{M}_0$, elementarily equivalent to $\mathfrak{N}$ (so that $\mathfrak{M}_0$ and $\mathfrak{N}$ satisfy exactly the same sentences) but not isomorphic to $\mathfrak{N}$.

PROOF.   We will construct $\mathfrak{M}_0$ by using the compactness theorem. Expand the language by adding a new constant symbol c. Let

$$\Sigma = \{\mathbf{0} < c, \mathbf{S0} < c, \mathbf{SS0} < c, \ldots\}.$$

We claim that $\Sigma \cup \text{Th}\,\mathfrak{N}$ has a model. For consider a finite subset. That finite subset is true in

$$\mathfrak{N}_k = (\mathbb{N}; 0, S, <, +, \cdot, k)$$

(where $k = c^{\mathfrak{N}_k}$) for some large k. So by the compactness theorem $\Sigma \cup \text{Th}\,\mathfrak{N}$ has a model.

By the Löwenheim–Skolem theorem, $\Sigma \cup \text{Th}\,\mathfrak{N}$ has a countable model

$$\mathfrak{M} = \left(|\mathfrak{M}|; \mathbf{0}^{\mathfrak{M}}, \mathbf{S}^{\mathfrak{M}}, <^{\mathfrak{M}}, +^{\mathfrak{M}}, \cdot^{\mathfrak{M}}, c^{\mathfrak{M}}\right).$$

Let $\mathfrak{M}_0$ be the restriction of $\mathfrak{M}$ to the original language:

$$\mathfrak{M}_0 = \left(|\mathfrak{M}|; \mathbf{0}^{\mathfrak{M}}, \mathbf{S}^{\mathfrak{M}}, <^{\mathfrak{M}}, +^{\mathfrak{M}}, \cdot^{\mathfrak{M}}\right).$$

Since $\mathfrak{M}_0$ is a model of $\text{Th}\,\mathfrak{N}$, we have $\mathfrak{M}_0 \equiv \mathfrak{N}$:

$$\models_{\mathfrak{N}} \sigma \;\Rightarrow\; \sigma \in \text{Th}\,\mathfrak{N} \;\Rightarrow\; \models_{\mathfrak{M}_0} \sigma$$
$$\not\models_{\mathfrak{N}} \sigma \;\Rightarrow\; \neg\sigma \in \text{Th}\,\mathfrak{N} \;\Rightarrow\; \models_{\mathfrak{M}_0} \neg\sigma \;\Rightarrow\; \not\models_{\mathfrak{M}_0} \sigma.$$

We leave it to the reader to verify that $\mathfrak{M}_0$ is not isomorphic to $\mathfrak{N}$. ($|\mathfrak{M}_0|$ contains the "infinite" number $c^{\mathfrak{M}}$.)   ⊣

What about uncountable[1] languages? Assume that in the proof of the completeness theorem we started with a set Γ in a language of cardinality λ. We claim that in this case, the structure $\mathfrak{A}/E$ we made has cardinality $\leq\lambda$.

$\mathfrak{A}/E$ was constructed from that preliminary structure $\mathfrak{A}$. The universe of $\mathfrak{A}$ was the set of all terms in the language obtained by adding λ new constant symbols. So the augmented language still had cardinality λ. So (by Theorem 0D) the set of all expressions (and hence the set of all terms) had cardinality $\leq\lambda$. (In fact, because we had at least the λ new constant symbols, the set of terms had cardinality exactly λ.)

The universe of $\mathfrak{A}/E$ consisted of equivalence classes of members of $\mathfrak{A}$, so card $|\mathfrak{A}/E| \leq$ card $|\mathfrak{A}|$. (We can map $|\mathfrak{A}/E|$ one-to-one into $|\mathfrak{A}|$ by assigning to each equivalence class some chosen member, but now we may need the axiom of choice.) Thus, when the smoke had cleared, Γ was satisfied in a structure $\mathfrak{A}/E$ of cardinality $\leq\lambda$.

LÖWENHEIM–SKOLEM THEOREM   (a) Let Γ be a satisfiable set of formulas in a language of cardinality λ. Then Γ is satisfiable in some structure of size $\leq\lambda$.

(b) Let Σ be a set of sentences in a language of cardinality λ. If Σ has any model, then it has a model of cardinality $\leq\lambda$.

---

[1] The reader who wishes to avoid uncountable cardinals is advised to skip to the subsection *Theories*.

The earlier version of the Löwenheim–Skolem theorem is a special case of this version, wherein $\lambda = \aleph_0$.

Suppose that we have an uncountable structure $\mathfrak{A}$ for a countable language. By the Löwenheim–Skolem theorem (applied to Th $\mathfrak{A}$) there is a countable $\mathfrak{B}$ that is a model of Th $\mathfrak{A}$, and hence $\mathfrak{A} \equiv \mathfrak{B}$, as noted previously.

Conversely, suppose that we start with a countable structure $\mathfrak{B}$. Is there an uncountable $\mathfrak{A}$ such that $\mathfrak{A} \equiv \mathfrak{B}$? If $\mathfrak{B}$ is finite (and the language includes equality), then this is impossible. But if $\mathfrak{B}$ is infinite, then there will be such an $\mathfrak{A}$, by the following "upward and downward Löwenheim–Skolem theorem." The upward part is due to Tarski, whence the "T" of "LST."

LST Theorem   Let Γ be a set of formulas in a language of cardinality λ, and assume that Γ is satisfiable in some infinite structure. Then for every cardinal $\kappa \geq \lambda$, there is a structure of cardinality κ in which Γ is satisfiable.

Proof.   Let $\mathfrak{A}$ be the infinite structure in which Γ is satisfiable. Expand the language by adding a set C of κ new constant symbols. Let

$$\Sigma = \{c_1 \neq c_2 \mid c_1, c_2 \text{ distinct members of } C\}.$$

Then every finite subset of $\Sigma \cup \Gamma$ is satisfiable in the structure $\mathfrak{A}$, expanded to assign distinct objects to the finitely many new constant symbols in the subset. (Since $\mathfrak{A}$ is infinite, there is room to accommodate any finite number of these.) So by compactness $\Sigma \cup \Gamma$ is satisfiable, and by the Löwenheim–Skolem theorem it is satisfiable in a structure $\mathfrak{B}$ of cardinality $\leq \kappa$. (The expanded language has cardinality $\lambda + \kappa = \kappa$.) But any model of Σ clearly has cardinality $\geq \kappa$. So $\mathfrak{B}$ has cardinality κ; restrict $\mathfrak{B}$ to the original language.   ⊣

Corollary 26F   (a) Let Σ be a set of sentences in a countable language. If Σ has some infinite model, then Σ has models of every infinite cardinality.

(b) Let $\mathfrak{A}$ be an infinite structure for a countable language. Then for any infinite cardinal λ, there is a structure $\mathfrak{B}$ of cardinality λ such that $\mathfrak{B} \equiv \mathfrak{A}$.

Proof.   (a) Take $\Gamma = \Sigma$, $\lambda = \aleph_0$ in the theorem.
(b) Take $\Sigma =$ Th $\mathfrak{A}$ in part (a).   ⊣

Consider a set Σ of sentences, to be thought of as nonlogical axioms. (For example, Σ might be a set of axioms for set theory or a set of axioms for number theory.) Call Σ *categorical* iff any two models of Σ are isomorphic. The above corollary implies that if Σ has any

infinite models, then Σ is not categorical. There is, for example, no set of sentences whose models are exactly the structures isomorphic to $(\mathbb{N}; 0, S, +, \cdot)$. This is indicative of a limitation in the expressiveness of first-order languages. (As will be seen in Section 4.1, there are categorical second-order sentences. But second-order sentences are peculiar objects, obtained at the cost of holding the notion of *subset* fixed, immune from interpretation by structures.)

Theories

We define a *theory* to be a set of sentences closed under logical implication. That is, T is a theory iff T is a set of sentences such that for any sentence σ of the language,

$$T \models \sigma \;\Rightarrow\; \sigma \in T.$$

(Note that we admit only sentences, not formulas, with free variables.)

For example, there is always a smallest theory, consisting of the valid sentences of the language. At the other extreme there is the theory consisting of all the sentences of the language; it is the only unsatisfiable theory.

For a class $\mathcal{K}$ of structures (for the language), define the *theory* of $\mathcal{K}$ (written Th $\mathcal{K}$) by the equation

$$\text{Th}\,\mathcal{K} = \{\sigma \mid \sigma \text{ is true in every member of } \mathcal{K}\}.$$

(This concept arose previously in the special case $\mathcal{K} = \{\mathfrak{A}\}$.)

THEOREM 26G   Th $\mathcal{K}$ is indeed a theory.

PROOF.   Any member of $\mathcal{K}$ is a model of Th $\mathcal{K}$. Thus if σ is true in every model of Th $\mathcal{K}$, then it is true in every member of $\mathcal{K}$. Whence it belongs to Th $\mathcal{K}$.   ⊣

For example, if the parameters of the language are $\forall$, $\mathbf{0}$, $\mathbf{1}$, $+$, and $\cdot$, and $\mathcal{F}$ is the class of all fields, then Th $\mathcal{F}$, the theory of fields, is simply the set of all sentences of the language which are true in all fields. If $\mathcal{F}_0$ is the class of fields of characteristic 0, then Th $\mathcal{F}_0$ is the theory of fields of characteristic 0.

Recall that for a set Σ of sentences, we defined Mod Σ to be the class of all models of Σ. Th Mod Σ is then the set of all sentences which are true in all models of Σ. But this is just the set of all sentences logically implied by Σ. Call this set the set of *consequences* of Σ, Cn Σ. Thus

$$\text{Cn}\,\Sigma = \{\sigma \mid \Sigma \models \sigma\}$$
$$= \text{Th Mod}\,\Sigma.$$

For example, set theory is the set of consequences of a certain set of sentences, known, unsurprisingly, as axioms for set theory. A set T of sentences is a theory iff $T = \text{Cn}\,T$.

A theory T is said to be *complete* iff for every sentence σ, either $\sigma \in T$ or $(\neg \sigma) \in T$. For example, for any one structure $\mathfrak{A}$, $\mathrm{Th}\{\mathfrak{A}\}$ (written, as before, "$\mathrm{Th}\,\mathfrak{A}$") is always a complete theory. In fact, it is clear upon reflection that $\mathrm{Th}\,\mathcal{K}$ is a complete theory iff any two members of $\mathcal{K}$ are elementarily equivalent. And a theory T is complete iff any two models of T are elementarily equivalent.

For example, the theory of fields is not complete, since the sentences

$$1 + 1 = 0,$$
$$\exists x \, x \cdot x = 1 + 1$$

are true in some fields but false in others. The theory of algebraically closed fields of characteristic 0 is complete, but this is by no means obvious. (See Theorem 26J.)

*Definition.   A theory T is *axiomatizable* iff there is a decidable set Σ of sentences such that $T = \mathrm{Cn}\,\Sigma$.

Definition.   A theory T is *finitely axiomatizable* iff $T = \mathrm{Cn}\,\Sigma$ for some finite set Σ of sentences.

In the latter case we have $T = \mathrm{Cn}\{\sigma\}$ (written "$T = \mathrm{Cn}\,\sigma$"), where σ is the conjunction of the finitely many members of Σ. For example, the theory of fields is finitely axiomatizable. For the class $\mathcal{F}$ of fields is $\mathrm{Mod}\,\Phi$, where Φ is the finite set of field axioms. And the theory of fields is $\mathrm{Th}\,\mathrm{Mod}\,\Phi = \mathrm{Cn}\,\Phi$.

The theory of fields of characteristic 0 is axiomatizable, being $\mathrm{Cn}\,\Phi_0$, where Φ_0 consists of the (finitely many) field axioms together with the infinitely many sentences:

$$1 + 1 \neq 0,$$
$$1 + 1 + 1 \neq 0,$$
$$\cdots$$

This theory is not finitely axiomatizable. To prove this, first note that no finite subset of Φ_0 has the entire theory as its set of consequences. (For that finite subset would be true in some field of very large characteristic.) Then apply the following:

Theorem 26H   If $\mathrm{Cn}\,\Sigma$ is finitely axiomatizable, then there is a finite $\Sigma_0 \subseteq \Sigma$ such that $\mathrm{Cn}\,\Sigma_0 = \mathrm{Cn}\,\Sigma$.

Proof.   Say that $\mathrm{Cn}\,\Sigma$ is finitely axiomatizable; then $\mathrm{Cn}\,\Sigma = \mathrm{Cn}\,\tau$ for some one sentence τ. In general $\tau \notin \Sigma$, but at least $\Sigma \models \tau$. ($\tau \in \mathrm{Cn}\,\tau = \mathrm{Cn}\,\Sigma$.) By the compactness theorem there is a finite $\Sigma_0 \subseteq \Sigma$ such that $\Sigma_0 \models \tau$. Then

$$\mathrm{Cn}\,\tau \subseteq \mathrm{Cn}\,\Sigma_0 \subseteq \mathrm{Cn}\,\Sigma,$$

whence equality holds. $\dashv$

We can now restate Corollaries 25F and 25G in the present terminology:

*COROLLARY 26I   (a) An axiomatizable theory (in a reasonable language) is effectively enumerable.

(b) A complete axiomatizable theory (in a reasonable language) is decidable.

We can represent the relationships among these concepts by means of a diagram (in which we have included the results of Exercise 6):
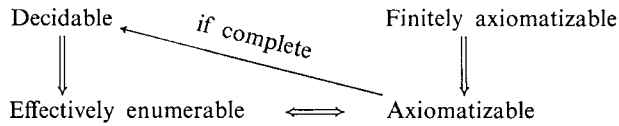
For example, a theory that is given in axiomatic form (such as Zermelo–Fraenkel set theory, which is Cn A_{ZF} for a certain set A_{ZF}) is effectively enumerable. We will argue in Section 3.7 that set theory (if consistent) is not decidable and not complete. Number theory, the theory of the structure $(\mathbb{N}; 0, S, <, +, \cdot, E)$, is complete but is not effectively enumerable and hence not axiomatizable (Section 3.5).

We can use part (b) of the preceding corollary to establish the decidability of an axiomatizable theory, provided we can show that the theory in question is complete. This can sometimes be done by means of the Łoś–Vaught test for completeness.

Say that a theory T is $\aleph_0$-*categorical* iff all the infinite countable models of T are isomorphic. More generally, for a cardinal κ, say that T is κ-*categorical* iff all models of T having cardinality κ are isomorphic.

ŁOŚ–VAUGHT TEST (1954)   Let T be a theory in a countable language. Assume that T has no finite models.

(a) If T is $\aleph_0$-categorical, then T is complete.

(b) If T is κ-categorical for some infinite cardinal κ, then T is complete.

PROOF.   It suffices to show for any two models $\mathfrak{A}$ and $\mathfrak{B}$ of T that $\mathfrak{A} \equiv \mathfrak{B}$. Since $\mathfrak{A}$ and $\mathfrak{B}$ are infinite, there exist (by the LST theorem) structures $\mathfrak{A}' \equiv \mathfrak{A}$ and $\mathfrak{B}' \equiv \mathfrak{B}$ having cardinality κ. $\mathfrak{A}'$ is isomorphic to $\mathfrak{B}'$, so we have

$$\mathfrak{A} \equiv \mathfrak{A}' \cong \mathfrak{B}' \equiv \mathfrak{B}.$$

Thus $\mathfrak{A} \equiv \mathfrak{B}$.   ⊣

(If T is a theory in a language of cardinality λ, then we must demand that $\lambda \leq \kappa$.)

The converse to the Łoś–Vaught test is false. That is, there are complete theories which are not κ-categorical for any κ.

In Section 3.1 we will apply the Łoś–Vaught test to prove the decidability of the theory of the natural numbers with zero and successor. It can also be used to prove the decidability of the theory of the complex field. But this proof will utilize an excursion into algebra.

THEOREM 26J (a) The theory of algebraically closed fields of characteristic 0 is complete.

*(b) The theory of the complex field

$$\mathfrak{C} = (\mathbb{C}; 0, 1, +, \cdot)$$

is decidable.

PROOF. Let $\mathcal{A}$ be the class of algebraically closed fields of characteristic 0. Then $\mathcal{A} = \text{Mod}(\Phi_0 \cup \Gamma)$, where Φ_0 consists as before of the axioms for fields of characteristic 0, and Γ consists of the sentences

$\forall a \, \forall b \, \forall c(a \neq \mathbf{0} \rightarrow \exists x \, a \cdot x \cdot x + b \cdot x + c = \mathbf{0})$,
$\forall a \, \forall b \, \forall c \, \forall d(a \neq \mathbf{0} \rightarrow \exists x \, a \cdot x \cdot x \cdot x + b \cdot x \cdot x + c \cdot x + d = \mathbf{0})$,
. . . .

The set $\Phi_0 \cup \Gamma$ is decidable and $\text{Th}\,\mathcal{A} = \text{Cn}(\Phi_0 \cup \Gamma)$, so this theory is axiomatizable. Part (a) of the theorem asserts that the theory is also complete, whence it is decidable.

Part (b) follows from part (a). For we have $\mathfrak{C} \in \mathcal{A}$, whence $\text{Th}\,\mathcal{A} \subseteq \text{Th}\,\mathfrak{C}$. The completeness of $\text{Th}\,\mathcal{A}$ implies that equality holds; see Exercise 2.

To prove part (a), we apply the Łoś–Vaught test. The models of $\text{Th}\,\mathcal{A}$ are exactly the members of $\mathcal{A}$. These are all infinite. We further claim that $\text{Th}\,\mathcal{A}$ is categorical in any uncountable cardinality. This is equivalent to saying that any two algebraically closed fields of characteristic 0 having the same uncountable cardinality are isomorphic.

This last assertion is a known result of algebra. We will sketch the proof, for the interest of those readers familiar with this topic. Any field $\mathfrak{F}$ is obtainable in the following way: (1) One begins with the prime subfield, which is determined to within isomorphism by the characteristic of $\mathfrak{F}$. (2) One takes a transcendental extension, determined within isomorphism by the cardinality of the transcendence basis, i.e., by the transcendence degree of $\mathfrak{F}$ (over its prime subfield). (3) One finally takes some algebraic extension. We thus have a theorem of Steinitz: Two algebraically closed fields are isomorphic iff they have the same characteristic and the same transcendence degree.

If the transcendence degree of an infinite field $\mathfrak{F}$ is κ, then the cardinality of $\mathfrak{F}$ is the larger of κ and $\aleph_0$. Hence for an uncountable

field, the cardinality equals the transcendence degree. So we conclude from Steinitz's theorem that two algebraically closed fields having the same characteristic and the same uncountable cardinality are isomorphic.   ⊣

The theory of the real field

$$(\mathbb{R}; 0, 1, +, \cdot)$$

is also decidable. But this result (which is due to Tarski) is much deeper than the above theorem. The theory of the real field is not categorical in any infinite cardinality, so the Łoś–Vaught test cannot be applied.

As a final application, we can show that the ordering of the rationals is elementarily equivalent to the ordering of the reals,

$$(\mathbb{Q}; <_Q) \equiv (\mathbb{R}; <_R),$$

where $\mathbb{Q}$ and $\mathbb{R}$ are the rationals and reals, respectively, and $<_Q$ and $<_R$ are the corresponding orderings. To show elementary equivalence, we show that both are models of some complete theory (which then must coincide with the theory of each structure). The key fact is provided by a theorem of Cantor: Any two countable dense linear orderings without endpoints are isomorphic.

To give the details, we must back up a little. The language here has equality and the parameters $\forall$ and $<$. Let δ be the conjunction of the following sentences:

1. Ordering axioms (trichotomy and transitivity):

$$\forall x \, \forall y (x < y \lor x = y \lor y < x),$$
$$\forall x \, \forall y (x < y \rightarrow y \not< x),$$
$$\forall x \, \forall y \, \forall z (x < y \rightarrow y < z \rightarrow x < z).$$

2. Density:

$$\forall x \, \forall y (x < y \rightarrow \exists z (x < z < y)).$$

3. No endpoints:

$$\forall x \, \exists y \, \exists z (y < x < z).$$

The dense linear orderings without endpoints are, by definition, the structures for this language that are models of δ. It is clear that they are all infinite. Furthermore, we claim that the theory of these orderings, Cn δ, is $\aleph_0$-categorical. This is provided by the following fact.

THEOREM 26K (CANTOR)   Any countable model of δ is isomorphic to $(\mathbb{Q}, <_Q)$.

We leave the proof to Exercise 4.

We can now apply the Łoś–Vaught test to conclude that $\mathrm{Cn}\,\delta$ is complete. Hence any two models of δ are elementarily equivalent; in particular,

$$(\mathbb{Q}; <_Q) \equiv (\mathbb{R}; <_R).$$

We can also conclude that these structures have decidable theories.

Prenex Normal Form

It will at times be convenient to move all the quantifier symbols to the left of other symbols. For example,

$$\forall x(Ax \rightarrow \forall y\, Bxy)$$

is equivalent to

$$\forall x\, \forall y(Ax \rightarrow Bxy).$$

And

$$\forall x(Ax \rightarrow \exists y\, Bxy)$$

is equivalent to

$$\forall x\, \exists y(Ax \rightarrow Bxy).$$

Define a *prenex* formula to be one of the form (for some $n \geq 0$)

$$Q_1 x_1 \cdots Q_n x_n\, \alpha,$$

where Q_i is $\forall$ or $\exists$ and α is quantifier-free.

PRENEX NORMAL FORM THEOREM   For any formula, we can find a logically equivalent prenex formula.

PROOF.   We will make use of the following quantifier manipulation rules.

Q1a. $\neg \forall x\, \alpha \models\!\mid \exists x\, \neg \alpha.$
Q1b. $\neg \exists x\, \alpha \models\!\mid \forall x\, \neg \alpha.$
Q2a. $(\alpha \rightarrow \forall x\, \beta) \models\!\mid \forall x(\alpha \rightarrow \beta)$ for x not free in α.
Q2b. $(\alpha \rightarrow \exists x\, \beta) \models\!\mid \exists x(\alpha \rightarrow \beta)$ for x not free in α.
Q3a. $(\forall x\, \alpha \rightarrow \beta) \models\!\mid \exists x(\alpha \rightarrow \beta)$ for x not free in β.
Q3b. $(\exists x\, \alpha \rightarrow \beta) \models\!\mid \forall x(\alpha \rightarrow \beta)$ for x not free in β.

Q1 is clear; for the others see the examples in Section 2.4 and Exercise 8 there.

We now show by induction that every formula has an equivalent prenex formula.

1. For atomic formulas this is vacuous, as any quantifier-free formula is trivially a prenex formula.
2. If α is equivalent to the prenex α', then $\forall x\, \alpha$ is equivalent to the prenex $\forall x\, \alpha'$.

3. If α is equivalent to the prenex α', then $\neg\,\alpha$ is equivalent to $\neg\,\alpha'$. Apply Q1 to $\neg\,\alpha'$ to obtain a prenex formula; for example,

$$\neg\,\forall x\,\exists\,y\,\exists z\,\beta \models\dashv \exists x\,\forall y\,\forall z\,\neg\,\beta.$$

4. Finally we come to the case of $\alpha \rightarrow \beta$. By inductive hypothesis we have prenex formulas α' and β' equivalent to α and β, respectively. By our theorems on alphabetic variants, we may further assume that any variable that occurs quantified in one of the formulas α' and β' does not occur at all in the other. We then use Q2 and Q3 to obtain a prenex formula equivalent to $\alpha' \rightarrow \beta'$ (and hence to $\alpha \rightarrow \beta$). Observe that there is some latitude in the order in which the rules Q2 and Q3 are applied. For example,

$$\forall x\,\exists\,y\,\varphi \rightarrow \exists\,u\,\psi$$

(where x and y do not occur in ψ, u does not occur in φ) is equivalent to any of the following:

$$\exists x\,\forall y\,\exists u(\varphi \rightarrow \psi),$$
$$\exists x\,\exists u\,\forall y(\varphi \rightarrow \psi),$$
$$\exists u\,\exists x\,\forall y(\varphi \rightarrow \psi).$$

$$\dashv$$

Retrospectus

At the beginning of this book it was stated that symbolic logic is a mathematical model of deductive thought. This is as good a time as any to reflect on that statement, in the light of the material treated thus far.

As a first example, consider a mathematician working in set theory. He uses a language with an equality symbol, a symbol $\in$ for membership, and numerous defined symbols ($\varnothing, \cup$, and others). In principle the defined symbols could be eliminated and any sentence replaced by an equivalent sentence in which the defined symbols did not appear. (In this connection see Section 2.7, where this topic is treated systematically.) He takes as primitive (or undefined) notions the concepts of set and membership. He adopts some set A_{ST} of axioms involving these concepts. He asserts that for certain sentences (his theorems), these sentences are true provided the axioms are true, regardless of what the undefined notions of set and membership actually mean. In support of these assertions he offers proofs, which are finitely long arguments intended to convince his colleagues of the correctness of the assertions.

In terms of first-order logic we can describe all this as follows: The language here is a first-order language with equality and a two-place predicate symbol $\in$. Thus $\forall$ and $\in$ are the only parameters open to interpretation. There is a certain set A_{ST} of sentences in this language

singled out as being the set of (nonlogical) axioms. Then certain other sentences are logical consequences of A_{ST}, i.e., are true in any model of A_{ST}. If τ is a consequence of A_{ST} (and only then), there is a deduction of τ from A_{ST}.

Next consider a more typical case of the hypothetical working mathematician, that of the algebraist or analyst. The algebraist uses axioms for (say) group theory, but he also employs some amount of set theory. Similarly, the analyst deals with sentences that involve both numbers and sets of numbers. In both cases it is generally recognized that one could, in principle, convert the assertions of algebra and analysis to assertions of set theory. And then the remarks of the preceding paragraph again apply.

The interest that symbolic logic holds for the mathematician is largely due to the accuracy with which it mirrors mathematical deductions. In the long run, it will surely be useful to understand the fundamental processes of doing mathematics.

There remains the question of the accuracy with which first-order logic mirrors nonmathematical deductive thought. Logic, symbolic and nonsymbolic, has always formed a traditional part of the philosophical study of the process by which people come to hold certain ideas. Nonmathematical examples to which first-order logic applies are provided by a vast array of frivolous situations. Lewis Carroll gave such examples, one of which inferred that babies cannot manage crocodiles from the three hypotheses: (1) Babies are illogical. (2) Nobody is despised who can manage a crocodile. (3) Illogical persons are despised.

But what of nonfrivolous situations? Here the applicability is obscured by the fact that we usually do not make explicit the assumptions we use in drawing conclusions. There are specific areas (in diverse fields such as physics, medicine, and law) where assumptions not only can be made explicit but are being made explicit. In some cases it appears that less than the full versatility of first-order logic is required to formalize the real-life deductions. Possibly in other cases — ranging from everyday life to quantum mechanics — more features may be necessary.

Exercises

1. Show that the following sentences are finitely valid (i.e., that they are true in every finite structure):
   (a) $\exists x \, \exists y \, \exists z [(Pxfx \rightarrow Pxx) \vee (Pxy \wedge Pyz \wedge \neg Pxz)]$
   (b) $\exists x \, \forall y \, \exists z [(Qzx \rightarrow Qzy) \rightarrow (Qxy \rightarrow Qxx)]$
   Suggestion: Show that any model of the negation must be infinite.

2. Let T_1 and T_2 be theories (in the same language) such that (i) $T_1 \subseteq T_2$, (ii) T_1 is complete, and (iii) T_2 is satisfiable. Show that $T_1 = T_2$.

3. Establish the following facts:
   (a) $\Sigma_1 \subseteq \Sigma_2 \Rightarrow \text{Mod}\,\Sigma_2 \subseteq \text{Mod}\,\Sigma_1$.

   $$\mathcal{K}_1 \subseteq \mathcal{K}_2 \Rightarrow \text{Th}\,\mathcal{K}_2 \subseteq \text{Th}\,\mathcal{K}_1.$$

   (b) $\Sigma \subseteq \text{Th Mod}\,\Sigma$ and $\mathcal{K} \subseteq \text{Mod Th}\,\mathcal{K}$.
   (c) $\text{Mod}\,\Sigma = \text{Mod Th Mod}\,\Sigma$ and $\text{Th}\,\mathcal{K} = \text{Th Mod Th}\mathcal{K}$. (Part (c) follows from (a) and (b).)

4. Prove that any two countable dense linear orderings without endpoints are isomorphic (Theorem 26K). *Suggestions*: Let $\mathfrak{A}$ and $\mathfrak{B}$ be such structures with $|\mathfrak{A}| = \{a_0, a_1, \ldots\}$ and $|\mathfrak{B}| = \{b_0, b_1, \ldots\}$. Construct an isomorphism in stages; at stage $2n$ be sure a_n is paired with some suitable b_j, and at stage $2n + 1$ be sure b_n is paired with some suitable a_i.

5. Find prenex formulas equivalent to the following.
   (a) $(\exists x\, Ax \wedge \exists x\, Bx) \rightarrow Cx$.
   (b) $\forall x\, Ax \leftrightarrow \exists x\, Bx$.

*6. Prove the converse to part (a) of Corollary 26I: An effectively enumerable theory (in a reasonable language) is axiomatizable. *Suggestion*: The set $\{\sigma_0, \sigma_1, \sigma_2, \ldots\}$ is equivalent (in the sense of having the same models) to the set $\{\sigma_0, \sigma_0 \wedge \sigma_1, \sigma_0 \wedge \sigma_1 \wedge \sigma_2, \ldots\}$.

7. Consider a language with a two-place predicate symbol $<$, and let $\mathfrak{N} = (\mathbb{N}; <)$ be the structure consisting of the natural numbers with their usual ordering. Show that there is some $\mathfrak{A}$ elementarily equivalent to $\mathfrak{N}$ such that $<^{\mathfrak{A}}$ has a descending chain. (That is, there must be $a_0, a_1, \ldots$ in $|\mathfrak{A}|$ such that $\langle a_{i+1}, a_i \rangle \in <^{\mathfrak{A}}$ for all i.) *Suggestion*: Apply the compactness theorem.

   Remark: The point of this exercise is to demonstrate that in this language, one can *not* express, "There is no descending chain."

8. Assume that σ is true in all infinite models of a theory T. Show that there is a finite number k such that σ is true in all models $\mathfrak{A}$ of T for which $|\mathfrak{A}|$ has k or more elements.

9. Say that a set Σ of sentences has the *finite model property* iff each member σ of Σ, if it has any models at all, has a finite model. Assume that Σ is a set of sentences in a finite language (i.e., a language with finitely many parameters) and that Σ has the finite model property. Give an effective procedure that, given any member σ of Σ, will decide whether or not σ has any models. *Suggestion*: Is the set of such sentences effectively enumerable? Is its complement effectively enumerable?

10. Assume we have a finite language without function symbols.
    (a) Show that the set of satisfiable $\exists_2$ sentences is decidable. (See Exercise 19 in Section 2.2 for terminology and background.) *Suggestion*: Apply the preceding exercise.

(b) Show that the set of valid $\forall_2$ sentences is decidable. (A $\forall_2$ formula is one of the form $\forall x_1 \cdots \forall x_m \exists y_1 \cdots \exists y_n \, \theta$ where θ is quantifier-free.)

Remarks: In first-order logic, the "decision problem" (Entscheidungsproblem) is the problem of deciding, given a formula, whether or not it is valid. By Church's theorem (Section 3.5), this problem in general is unsolvable. This exercise gives a solvable subcase of the decision problem.

SECTION 2.7

Interpretations Between Theories[1]

In some cases a theory T_1 can be shown to be every bit as powerful as another theory T_0. This is certainly the case if the theories are in the same language and $T_0 \subseteq T_1$. But even if the theories are in different languages, there may exist a way of translating from one language to the other in such a way that members of T_0 are translated as members of T_1. This sort of situation will be examined in this section.

We will begin by discussing the topic of defined symbols. This topic, as well as having significant interest of its own, will serve as an example for the situation of the preceding paragraph, wherein T_0 is constructed from T_1 by adding a new defined symbol. If the definition is done properly, the original theory T_1 should in principle be just as strong as the new T_0. We will consider only the case of defined function symbols, since the case of defined predicate symbols presents, in comparison, no real difficulties.

Defining Functions

Frequently in mathematics it is useful to introduce definitions of new functions. For example, in set theory one defines the power-set operation $\mathcal{P}$ by a sentence like, "Let $\mathcal{P}x$ be the set whose members are the subsets of x." Or by a sentence in a formal language (here containing $\in$, $\subseteq$, and $\mathcal{P}$),

$$\forall v_1 \, \forall v_2 [\mathcal{P}v_1 = v_2 \leftrightarrow \forall u(u \in v_2 \leftrightarrow u \subseteq v_1)].$$

Now definitions are unlike theorems and unlike axioms. Unlike theorems, definitions are not things we prove. We just declare them by fiat. But unlike axioms, we do not expect definitions to add substantive information. A definition is expected to add to our convenience, not to our knowledge.

---

[1] The results of this section will be used only in the latter part of Section 3.7.

If this expectation is to be realized, the definition must be made in a reasonable way. As an example of a most unreasonable definition in number theory, suppose that we introduce a new function symbol by the "definition"

$$f(x) = y \quad \text{iff} \quad x < y.$$

(Or by the sentence in a formal language: $\forall\, v_1 \,\forall\, v_2 (f\, v_1 = v_2 \leftrightarrow v_1 < v_2)$.) Since we know that $1 < 2$, we see that $f(1) = 2$. But also $1 < 3$, so we obtain $f(1) = 3$. And so we come to the conclusion (which does not itself involve f) that $2 = 3$.

Obviously this definition of f was in some way very bad. It did not just make matters convenient for us; it enabled us to conclude that $2 = 3$, which we could not have done without the definition. The trouble was that the definition bestowed the name "$f(1)$" ambiguously upon many things (2 and 3 among them). Thus $f(1)$ was not "well defined." Names ought to designate unique objects.

In this subsection we want to consider conditions under which we can be assured that a definition will be satisfactory. To simplify the notation, we will consider only the definition of a one-place function symbol f, but the remarks will apply to n-place function symbols as well.

Consider then a theory T in a language not yet containing the one-place function symbol f. (For example, T might be the set of consequences of your favorite axioms for set theory.) We want to add f to the language, introducing it by the definition

$$\forall\, v_1 \,\forall\, v_2 [f\, v_1 = v_2 \leftrightarrow \varphi], \tag{δ}$$

where φ is a formula in the original language (i.e., a formula not containing f) in which only v_1 and v_2 may occur free.

THEOREM 27A In the above situation, the following are equivalent:

(a) (The definition is noncreative.) For any sentence σ in the smaller language, if

$$T; \delta \models \sigma$$

(in the augmented language), then already $T \models \sigma$.
(b) (f is well defined.) The sentence

$$\forall\, v_1 \,\exists!\, v_2 \,\varphi \tag{ε}$$

is in the theory T. (Here "$\exists!\, v_2 \,\varphi$" is an abbreviation for a longer formula; see Exercise 21 of Section 2.2.)

PROOF.   To obtain (a) $\Rightarrow$ (b), simply note that $\delta \models \varepsilon$. So by taking $\sigma = \varepsilon$ in part (a), we obtain $T \models \varepsilon$.

Conversely, assume that $T \models \varepsilon$. Let $\mathfrak{A}$ be a model of T. ($\mathfrak{A}$ is a structure for the original language.) For $d \in |\mathfrak{A}|$, let $F(d)$

be the unique $e \in |\mathfrak{A}|$ such that $\models_{\mathfrak{A}} \varphi[\![d, e]\!]$. (There is a unique such e because $\models_{\mathfrak{A}} \varepsilon$.) Let $(\mathfrak{A}, F)$ be the structure for the augmented language that agrees with $\mathfrak{A}$ on the original parameters and that assigns F to the symbol f. Then it is easy to see that $(\mathfrak{A}, F)$ is a model of δ. Furthermore, $\mathfrak{A}$ and $(\mathfrak{A}, F)$ satisfy the same sentences of the original language. In particular $(\mathfrak{A}, F)$ is a model of T. Hence

$$T; \delta \models \sigma \Rightarrow \models_{(\mathfrak{A}, F)} \sigma$$
$$\Rightarrow \models_{\mathfrak{A}} \sigma. \qquad \dashv$$

(This argument can be stated more briefly by using second-order logic. ε is logically equivalent to the sentence $\exists f \delta$.)

Interpretations

The basic idea is that it is possible for one theory to be just as strong (in a sense to be made precise) as another theory in another language. In considering two languages simultaneously, there is no point in allowing them to conflict; e.g., the negation symbol of one language should not be a predicate symbol in the other. We can eliminate such conflicts by assuming that each of the languages is obtained from a third parent language by deleting some parameters (and perhaps equality).

For example, axiomatic set theory is at least as strong as the theory of the natural numbers with zero and successor, i.e., the theory of $(\mathbb{N}; 0, S)$. Any sentence in the language of $(\mathbb{N}; 0, S)$ can be translated in a natural way into a sentence of set theory. (This translation is sketched briefly in Section 3.7.) If the original sentence was true in $(\mathbb{N}; 0, S)$, then the translation will be a consequence of the axioms of set theory. (This is not obvious; the proof uses facts to be developed in Section 3.1.)

Let us look more carefully at a second example. Consider on the one hand the theory of

$$(\mathbb{N}; 0, S)$$

in its language, and on the other hand the theory of

$$(\mathbb{Z}; +, \cdot)$$

in its language. (Here $\mathbb{Z}$ is the set of all integers, positive, negative, and zero.) We will shortly be in a position to claim that the second theory is as strong as the first. How might a sentence about the natural numbers $\mathbb{N}$ with 0 and S be translated into a sentence about the integers $\mathbb{Z}$ with addition and multiplication?

The first clue is provided by Lagrange's theorem from number theory: an integer is nonnegative iff it is the sum of four squares. Thus a quantifier $\forall x$ in the first language (where x is intended to range over $\mathbb{N}$)

can be replaced by

$$\forall x (\exists y_1 \exists y_2 \exists y_3 \exists y_4 \, x = y_1 \cdot y_1 + y_2 \cdot y_2 + y_3 \cdot y_3 + y_4 \cdot y_4 \rightarrow$$

in the second language.

The second clue is that $\{0\}$ and the successor function (viewed as a relation) are definable in $(\mathbb{Z}; +, \cdot)$. The set $\{0\}$ is defined by

$$v_1 + v_1 = v_1.$$

The successor relation (extended to $\mathbb{Z}$) is defined by

$$\forall z (z \cdot z = z \wedge z + z \neq z \rightarrow v_1 + z = v_2).$$

Thus the sentence about $(\mathbb{N}; 0, S)$

$$\forall x \, \mathbf{S}x \neq \mathbf{0}$$

can be translated into

$$\forall x [\exists y_1 \exists y_2 \exists y_3 \exists y_4 \, x = y_1 \cdot y_1 + y_2 \cdot y_2 + y_3 \cdot y_3 + y_4 \cdot y_4 \rightarrow$$
$$\neg \forall u (u + u = u \rightarrow \forall v (\forall z (z \cdot z = z \wedge z + z \neq z \rightarrow x + z$$
$$= v) \rightarrow v = u))].$$

So much for examples. For our general discussion it will be helpful to introduce the notation

$$\varphi(t) = \varphi_t^{v_1},$$
$$\varphi(t_1, t_2) = \left(\varphi_{t_1}^{v_1}\right)_{t_2}^{v_2},$$

and so forth. Thus $\varphi = \varphi(v_1) = \varphi(v_1, v_2)$. If we use "$\varphi(x)$" we will not worry too much about whether or not x is substitutable for v_1 in φ. If it is not, then we actually want $\varphi(x)$ to be $\psi_x^{v_1}$, where ψ is a suitable alphabetic variant of φ.

Assume now that we have the following general situation:

L_0 is a language. (A *language* can for all practical purposes be a set of parameters, possibly augmented by the equality symbol.)

T_1 is a theory in a (possibly different) language L_1, which includes equality.

DEFINITION.   An *interpretation* π of L_0 into T_1 is a function on the set of parameters of L_0 such that

  1. π assigns to $\forall$ a formula $\pi_\forall$ of L_1 in which at most v_1 occurs free, such that

  (i) $T_1 \models \exists v_1 \pi_\forall.$

  (The idea is that in any model of T_1, the formula $\pi_\forall$ should define a nonempty set to be used as the universe of an L_0-structure.)

  2. π assigns to each n-place predicate parameter P a formula π_P of L_1 in which at most the variables $v_1, \dots v_n$ occur free.

3. π assigns to each n-place function symbol f a formula π_f of L_1, in which at most $v_1, \ldots, v_n, v_{n+1}$ occur free, such that

(ii) $\begin{aligned} T_1 \models \forall\, v_1 \cdots \forall\, v_n(\pi_\forall(v_1) \rightarrow \cdots \rightarrow \pi_\forall(v_n) \rightarrow \\ \exists\, x(\pi_\forall(x) \wedge \forall\, v_{n+1}(\pi_f(v_1, \ldots, v_{n+1}) \leftrightarrow v_{n+1} = x))). \end{aligned}$

(In English, this formula becomes, "For all $\vec{v}$ in the set defined by $\pi_\forall$, there is a unique x such that $\pi_f(\vec{v}, x)$, and furthermore x is in the set defined by $\pi_\forall$." The idea is to ensure that in any model of T_1, π_f defines a function on the universe defined by $\pi_\forall$. In the case of a constant symbol c, we have $n = 0$ and (ii) becomes

$$T_1 \models \exists\, x(\pi_\forall(x) \wedge \forall\, v_1(\pi_c(v_1) \leftrightarrow v_1 = x)).$$

In other words, π_c defines a singleton whose one member is also in the set defined by $\pi_\forall$.)

For example, if L_0 is the language of $(\mathbb{N}; 0, S)$ and T_1 is the theory of $(\mathbb{Z}; +, \cdot)$, then we have

$$\pi_\forall(x) = \exists\, y_1 \exists\, y_2 \exists\, y_3 \exists\, y_4\, x = y_1 \cdot y_1 + y_2 \cdot y_2 + y_3 \cdot y_3 + y_4 \cdot y_4,$$
$$\pi_0(x) = x + x = x,$$
$$\pi_S(x, y) = \forall_z (z \cdot z = z \wedge z + z \neq z \rightarrow x + z = y).$$

(We are here exploiting the fact that in $(\mathbb{Z}; +, \cdot)$ we can, in effect, define the structure $(\mathbb{N}; 0, S)$.)

If L_0 coincides with L_1, there is trivially the identity interpretation π, for which

$$\pi_\forall = v_1 = v_1,$$
$$\pi_P = P\, v_1 \cdots v_n,$$
$$\pi_f = f\, v_1 \cdots v_n = v_{n+1}.$$

The conditions (i) and (ii) are then met no matter what T_1 is.

Now assume that π is an interpretation and let $\mathfrak{B}$ be a model of T_1. There is a natural way to extract from $\mathfrak{B}$ a structure $^\pi\mathfrak{B}$ for L_0. Namely, let

$$|^\pi\mathfrak{B}| = \text{the set defined in } \mathfrak{B} \text{ by } \pi_\forall,$$
$$P^{\pi\mathfrak{B}} = \text{the relation defined in } \mathfrak{B} \text{ by } \pi_P, \text{ restricted to } |^\pi\mathfrak{B}|,$$
$$f^{\pi\mathfrak{B}}(a_1, \ldots, a_n) = \text{the unique } b \text{ such that } \models_\mathfrak{B} \pi_f[\![a_1, \ldots, a_n, b]\!],$$
$$\text{where } a_1, \ldots, a_n \text{ are in } |^\pi\mathfrak{B}|.$$

By condition (i) in the definition of interpretations, $|^\pi\mathfrak{B}| \neq \varnothing$. And, by condition (ii), the definition of $f^{\pi\mathfrak{B}}$ makes sense; i.e., there is a unique b meeting the above condition. Hence $^\pi\mathfrak{B}$ is indeed a structure for the language L_0.

Define the set $\pi^{-1}[T_1]$ of L_0-sentences by the equation

$$\pi^{-1}[T_1] = \text{Th}\{^\pi\mathfrak{B} \mid \mathfrak{B} \in \text{Mod } T_1\}$$
$$= \{\sigma \mid \sigma \text{ is an } L_0\text{-sentence true in every structure } ^\pi\mathfrak{B}$$
$$\text{obtainable from a model } \mathfrak{B} \text{ of } T_1\}.$$

This *is* a theory, as is $\mathrm{Th}\,\mathcal{K}$ for any class $\mathcal{K}$. It is a satisfiable theory iff T_1 is satisfiable.

> **EXAMPLE.**   Earlier in this section we had a theory T containing the sentence
>
> $$\forall\, v_1 \,\exists!v_2\,\varphi. \tag{ε}$$
>
> We augmented the language to a larger language L^+ that contained a function symbol f. The "definition" of f was provided by the L^+-sentence
>
> $$\forall\, v_1 \,\forall\, v_2(f\,v_1 = v_2 \leftrightarrow \varphi). \tag{δ}$$
>
> We showed that for a sentence σ in the original language of T, if $T;\delta \models \sigma$, then $T \models \sigma$.
>
> We have an interpretation π from L^+ into T. π is the identity interpretation on all parameters except f. The formula π_f is φ. The fact that $T \models \varepsilon$ is just what we need to verify that π is indeed an interpretation. For any model $\mathfrak{A}$ of T, $^\pi\mathfrak{A}$ is a structure previously called $(\mathfrak{A}, F)$; it is a model of $T;\delta$.
>
> We claim that
>
> $$\pi^{-1}[T] = \mathrm{Cn}(T;\delta).$$
>
> First observe that any model $\mathfrak{B}$ of $T;\delta$ equals $^\pi\mathfrak{B}$, where $\mathfrak{A}$ is the restriction of $\mathfrak{B}$ to the language of T. Hence for an L^+-sentence σ,
>
> $$\begin{aligned} \sigma \in \pi^{-1}[T] &\Leftrightarrow \models_{^\pi\mathfrak{A}} \sigma &&\text{for every model } \mathfrak{A} \text{ of } T \\ &\Leftrightarrow \models_{\mathfrak{B}} \sigma &&\text{for every model } \mathfrak{B} \text{ of } T;\delta \\ &\Leftrightarrow T;\delta \models \sigma. \end{aligned}$$

Syntactical Translation

In the preceding subsection on interpretations we talked about arbitrary models and such. But the reader may already have noticed that there is a much more down-to-earth thing to be said about an interpretation π of L_0 into T_1. Briefly: We can, given a formula φ of L_0, find a formula φ^π in L_1 which in some sense corresponds exactly to φ. We define φ^π by recursion on φ.

First, consider an atomic formula α of L_0. For example, if α is

$$P f g x,$$

then α is logically equivalent to

$$\forall\, y(gx = y \rightarrow \forall\, z(fy = z \rightarrow Pz)).$$

And we can take for α^π the L_1-formula

$$\forall\, y(\pi_g(x, y) \rightarrow \forall\, z(\pi_f(y, z) \rightarrow \pi_P(z))).$$

In general, scan an atomic formula α from right to left. The rightmost place at which a function symbol occurs will initiate a segment of the form $g x_1 \cdots x_n$, for some n-place g. (In the example $n = 1$.) Replace this by some new variable y, and prefix $\forall\, y(\pi_g(x_1, \ldots, x_n, y)) \rightarrow$. Continue to the next place at which a function symbol occurs. Finally, replace the predicate symbol P (if a parameter) by π_P (with the correct variables).

The definition of α^π can be stated more carefully by using recursion on the number of places at which function symbols occur in α. If that number is zero, then α is $P x_1 \cdots x_n$ and α^π is $\pi_P(x_1, \ldots, x_n)$. Otherwise, take the rightmost place at which a function symbol g occurs. If g is an n-place symbol, then that place initiates a segment $g x_1 \cdots x_n$. Replace this segment by some new variable y, obtaining a formula we can call $\alpha_y^{g x_1 \cdots x_n}$. Then α^π is

$$\forall\, y(\pi_g(x_1, \ldots, x_n, y) \rightarrow (\alpha_y^{g x_1 \cdots x_n})^\pi).$$

For example,

$$
\begin{aligned}
(P f g x)^\pi &= \forall\, y(\pi_g(x, y) \rightarrow (P f y)^\pi) \\
&= \forall\, y(\pi_g(x, y) \rightarrow \forall\, z(\pi_f(y, z) \rightarrow (P z)^\pi)) \\
&= \forall\, y(\pi_g(x, y) \rightarrow \forall\, z(\pi_f(y, z) \rightarrow \pi_P(z))).
\end{aligned}
$$

The interpretation of a nonatomic formula is defined in the obvious way. $(\neg\, \varphi)^\pi$ is $(\neg\, \varphi^\pi)$, $(\varphi \rightarrow \psi)^\pi$ is $(\varphi^\pi \rightarrow \psi^\pi)$, and $(\forall\, x\, \varphi)^\pi$ is $\forall\, x(\pi_\forall(x) \rightarrow \varphi^\pi)$. (Thus the quantifiers are "relativized" to $\pi_\forall$.)

The sense in which φ^π "says the same thing" as φ is made precise in the following basic lemma.

LEMMA 27B   Let π be an interpretation of L_0 into T_1, and let $\mathfrak{B}$ be a model of T_1. For any formula φ of L_0 and any map s of the variables into $|^\pi\mathfrak{B}|$,

$$\models_{\pi\mathfrak{B}} \varphi[s] \quad \text{iff} \quad \models_{\mathfrak{B}} \varphi^\pi[s].$$

This is not a deep fact. It just says that φ^π was defined correctly.

PROOF.   We use induction on φ, but only the case of an atomic formula α is nontrivial. For α, we use induction on the number of places at which function symbols occur. It is easy if that number is zero. Otherwise,

$$\alpha^\pi = \forall\, y(\pi_g(x, y) \rightarrow \beta^\pi),$$

where $\beta_{gx}^y = \alpha$. (We have quietly assumed that g is a one-place symbol; the notation is bad enough already.) Let

$$
\begin{aligned}
b &= \text{the unique } b \text{ such that } \models_{\mathfrak{B}} \pi_g[\![s(x), b]\!] \\
&= g^{\pi\mathfrak{B}}(s(x)).
\end{aligned}
$$

Then

$$\models_{\mathfrak{B}} \alpha^{\pi}[s] \Leftrightarrow \models_{\mathfrak{B}} \beta^{\pi}[s(y\,|\,b)]$$

$$\Leftrightarrow \models_{\pi\mathfrak{B}} \beta[s(y\,|\,b)] \quad \text{by the inductive hypothesis}$$

$$\Leftrightarrow \models_{\pi\mathfrak{B}} \beta_{gx}^{y}[s] \quad\quad \text{by the substitution lemma}$$

$$\Leftrightarrow \models_{\pi\mathfrak{B}} \alpha[s].$$

⊣

The following corollary justifies our choice of notation for $\pi^{-1}[T_1]$.

COROLLARY 27C   For a sentence σ of L_0,

$$\sigma \in \pi^{-1}[T_1] \quad \text{iff} \quad \sigma^{\pi} \in T_1.$$

PROOF.   Recall that by definition

$$\sigma \in \pi^{-1}[T_1] \Leftrightarrow \text{for every model } \mathfrak{B} \text{ of } T_1, \models_{\pi\mathfrak{B}} \sigma$$

$$\Leftrightarrow \text{for every model } \mathfrak{B} \text{ of } T_1, \models_{\mathfrak{B}} \sigma^{\pi} \text{ by Lemma 27B}$$

$$\Leftrightarrow T_1 \models \sigma^{\pi}.$$

⊣

DEFINITION.   An *interpretation* π of a theory T_0 into a theory T_1 is an interpretation π of the language of T_0 into T_1 such that

$$T_0 \subseteq \pi^{-1}[T_1].$$

In other words, it is necessary that for an L_0-sentence σ,

$$\sigma \in T_0 \Rightarrow \sigma^{\pi} \in T_1.$$

$\pi^{-1}[T_1]$ is the largest theory that π interprets into T_1. If $T_0 = \pi^{-1}[T_1]$, then we have

$$\sigma \in T_0 \Leftrightarrow \sigma^{\pi} \in T_1.$$

In this case π is said to be a *faithful* interpretation of T_0 into T_1.

To return to an earlier example, consider the structures $(\mathbb{N}; 0, S)$ and $(\mathbb{Z}; +, \cdot)$. We had an interpretation π into $\mathrm{Th}(\mathbb{Z}; +, \cdot)$, where

$$\pi_{\forall}(x) = \exists\, y_1 \, \exists\, y_2 \, \exists\, y_3 \, \exists\, y_4\, x = y_1 \cdot y_1 + y_2 \cdot y_2 + y_3 \cdot y_3 + y_4 \cdot y_4,$$

$$\pi_0(x) = x + x = x,$$

$$\pi_S(x, y) = \forall\, z(z \cdot z = z \wedge z + z \neq z \rightarrow x + z = y).$$

We now claim that π is a faithful interpretation of $\mathrm{Th}(\mathbb{N}; 0, S)$ into $\mathrm{Th}(\mathbb{Z}; +, \cdot)$. For in this case, $^{\pi}(\mathbb{Z}; +, \cdot)$ *is* the structure $(\mathbb{N}; 0, S)$. Hence

$$\models_{(\mathbb{N};0,S)} \sigma \iff \models \pi_{(\mathbb{Z};+,\cdot)} \sigma \iff \models_{(\mathbb{Z};+,\cdot)} \sigma^{\pi}.$$

In Chapter 3 we will be able to show that there is no interpretation of $\mathrm{Th}(\mathbb{Z}; +, \cdot)$ into $\mathrm{Th}(\mathbb{N}; 0, S)$. Thus the former theory is strictly stronger than the latter.

Finally, let us return to the situation with which we started this section. Assume that T is a theory containing the sentence ε, where

$\varepsilon = \forall v_1 \exists! v_2 \varphi;$
$\delta = \forall v_1 \forall v_2 (f v_1 = v_2 \leftrightarrow \varphi);$
$L^+ = $ the language obtained by adding the new function symbol f to the language of T;
$\pi = $ the interpretation of L^+ into T that is the identity interpretation on all parameters except f, and $\pi_f = \varphi$.

In fact, π is a faithful interpretation of $\mathrm{Cn}(T; \delta)$ into T, since, as noted previously,

$$\pi^{-1}[T] = \mathrm{Cn}(T; \delta).$$

We can now draw an additional conclusion; the definition is *eliminable*.

> **THEOREM 27D**   Assume that we have the situation described above. Then for any L^+-sentence σ we can find the sentence σ^π in the original language such that
>
> (a) $T; \delta \models (\sigma \leftrightarrow \sigma^\pi)$.
> (b) $T; \delta \models \sigma \iff T \models \sigma^\pi$.
> (c) If f does not occur in σ, then $\models (\sigma \leftrightarrow \sigma^\pi)$.

> PROOF.   Part (c) follows from the fact that π is the identity interpretation on all parameters except f. Part (b) restates that π is a faithful interpretation of $\mathrm{Cn}(T; \delta)$ into T. Since π is faithful, for (a) it suffices to show that
>
> $$T \models (\sigma \leftrightarrow \sigma^\pi)^\pi.$$
>
> This follows from (c), since $(\sigma \leftrightarrow \sigma^\pi)^\pi$ is $(\sigma^\pi \leftrightarrow \sigma^{\pi\pi})$, which is valid.   ⊣

Exercises

1. Assume that L_0 and L_1 are languages with the same parameters except that L_0 has an n-place function symbol f not in L_1 and L_1 has an $(n + 1)$-place predicate symbol P not in L_0. Show that for any L_0-theory T there is a faithful interpretation of T into some L_1-theory.

2. Let L_0 be the language with equality and the two-place function symbols $+$ and $\cdot$. Let L_1 be the same, but with three-place *predicate* symbols for addition and multiplication. Let $\mathfrak{N}_i = (\mathbb{N}; +, \cdot)$ be the structure for L_i consisting of the natural numbers with addition and multiplication $(i = 0, 1)$. Show that any relation definable by an L_0-formula in $\mathfrak{N}_0$ is also definable by an L_1-formula in $\mathfrak{N}_1$.

3. Show that an interpretation of a complete theory into a satisfiable theory is faithful.

SECTION 2.8

Nonstandard Analysis[1]

The differential and integral calculus was originally described by Leibniz and Newton in the seventeenth century in terms of quantities that were infinitely small yet nonzero. Newton used in his calculations a number o that, being infinitely small, could be multiplied by any finite number and still be negligible. But it was necessary to divide by o, so it had to be nonzero. Leibniz's dx was less than any assignable quantity, yet was nonzero.

These ideas were not easy to comprehend or to accept. Throughout the eighteenth century this business of working with infinitesimals was attacked (e.g., by Bishop Berkeley), distrusted (e.g., by D'Alembert), and used in enthusiastic experimentation (e.g., by Euler). While Euler was creating the mathematics that students now study in advanced calculus, he used infinitesimals in a loose, free-swinging manner that would not be tolerated in today's freshmen. Only in the nineteenth century were the foundations of calculus presented in the form now found in textbooks. The treatment of limits was then rigorous, and debate subsided.

In 1961 Abraham Robinson introduced a new method for treating limits, rescuing infinitesimals from their intellectual disrepute. This method combines the intuitive advantages of working with infinitely small quantities with modern standards of rigor. The basic idea is to utilize a nonstandard model of the theory of the real numbers.

Construction of $^*\mathfrak{R}$

We will use a very large first-order language. In addition to symbols for $+$, $\cdot$, and $<$ we might as well add symbols for the exponentiation and absolute-value functions. And since there is no good reason to stop there, we go all the way and include a symbol for *every* operation on the set $\mathbb{R}$ of reals. We do the same for every relation on $\mathbb{R}$. Thus we have the language with equality and the following parameters:

0. $\forall$, intended to mean "for all real numbers."
1. An n-place predicate symbol $\boldsymbol{P}_R$ for each n-ary relation R on $\mathbb{R}$.
2. A constant symbol $\boldsymbol{c}_r$ for each $r \in \mathbb{R}$.
3. An n-place function symbol $\boldsymbol{f}_F$ for each n-ary operation F on $\mathbb{R}$.

---

[1] This section may be omitted without loss of continuity.

For this language there is the standard structure $\mathfrak{R}$, with $|\mathfrak{R}| = \mathbb{R}$, $P_R^{\mathfrak{R}} = R$, $c_r^{\mathfrak{R}} = r$, and $f_F^{\mathfrak{R}} = F$. But now let us form a nonstandard structure, by using the compactness theorem. Let Γ be the set

$$\text{Th}\,\mathfrak{R} \cup \{c_r P_< v_1 \mid r \in \mathbb{R}\}.$$

(Here $c_r P_< v_1$ formalizes "r is less than v_1.") Any finite subset of Γ can be satisfied in $\mathfrak{R}$ by assigning to v_1 some large real number. Hence by the compactness theorem there is a structure $\mathfrak{A}$ and an element $a \in |\mathfrak{A}|$ such that Γ is satisfied in $\mathfrak{A}$ when v_1 is assigned a. Since $\mathfrak{A}$ is a model of Th $\mathfrak{R}$, we have $\mathfrak{A} \equiv \mathfrak{R}$. There is also an isomorphism h of $\mathfrak{R}$ into (but not onto) $\mathfrak{A}$ defined by

$$h(r) = c_r^{\mathfrak{A}}.$$

To check that this is indeed an isomorphism, we use the fact that $\mathfrak{A} \equiv \mathfrak{R}$. h is one-to-one, since for $r_1 \neq r_2$, the sentence $c_{r_1} \neq c_{r_2}$ holds in $\mathfrak{R}$ and hence in $\mathfrak{A}$. h preserves a binary relation $R(= P_R^{\mathfrak{R}})$ since for any r and s in $\mathbb{R}$,

$$\begin{aligned}\langle r, s \rangle \in P_R^{\mathfrak{R}} &\Leftrightarrow \models_{\mathfrak{R}} P_R c_r c_s \\ &\Leftrightarrow \models_{\mathfrak{A}} P_R c_r c_s \\ &\Leftrightarrow \langle c_r^{\mathfrak{A}}, c_s^{\mathfrak{A}} \rangle \in P_R^{\mathfrak{A}} \\ &\Leftrightarrow \langle h(r), h(s) \rangle \in P_R^{\mathfrak{A}}.\end{aligned}$$

A similar calculation applies to any n-ary relation. Next we must show that h preserves any function $F(= f_F^{\mathfrak{R}})$. Again for notational ease, suppose that F is a binary operation. Consider any r and s in $\mathbb{R}$, and let $t = F(r, s)$. Then

$$\begin{aligned}h\big(f_F^{\mathfrak{R}}(r, s)\big) &= h(F(r, s)) \\ &= h(t) \\ &= c_t^{\mathfrak{A}}.\end{aligned}$$

Now the sentence $c_t = f_F c_r c_s$ holds in $\mathfrak{R}$ and hence in $\mathfrak{A}$. Hence

$$\begin{aligned}c_t^{\mathfrak{A}} &= f_F^{\mathfrak{A}}\big(c_r^{\mathfrak{A}}, c_s^{\mathfrak{A}}\big) \\ &= f_F^{\mathfrak{A}}(h(r), h(s)).\end{aligned}$$

So h preserves f_F. For the constant symbols we have by the definition of h,

$$\begin{aligned}h\big(c_r^{\mathfrak{R}}\big) &= h(r) \\ &= c_r^{\mathfrak{A}}.\end{aligned}$$

Since we have an isomorphic copy of $\mathfrak{R}$ inside $\mathfrak{A}$, we can find another structure $*\mathfrak{R}$ isomorphic to $\mathfrak{A}$ such that $\mathfrak{R}$ is a substructure of $*\mathfrak{R}$. The idea is simply to replace in $\mathfrak{A}$ the point $c_r^{\mathfrak{A}}$ by the point r (provided that $|\mathfrak{A}| \cap \mathbb{R} = \varnothing$, as can always be arranged). For details, see Exercise 24 of Section 2.2. Since $*\mathfrak{R}$ is isomorphic to $\mathfrak{A}$, there is a point $b \in |*\mathfrak{R}|$ such that $*\mathfrak{R}$ satisfies Γ when v_1 is assigned b. In particular, $*\mathfrak{R} \equiv \mathfrak{R}$.

To go much further, we need a less cumbersome notation. We will use an asterisk to indicate passage from $\mathfrak{R}$ to $^*\mathfrak{R}$.

1. For each n-ary relation R on $\mathbb{R}$, let $^*R$ be the relation $P_R^{^*\mathfrak{R}}$ assigned to the symbol P_R by $^*\mathfrak{R}$. In particular, $\mathbb{R}$ is a unary relation on $\mathbb{R}$. Its image $^*\mathbb{R}$ equals the universe of $^*\mathfrak{R}$, since the sentence $\forall x P_{\mathbb{R}} x$ is true in $\mathfrak{R}$ and hence in $^*\mathfrak{R}$. Since $\mathfrak{R}$ is a substructure of $^*\mathfrak{R}$, we have that each relation R equals the restriction of $^*R$ to $\mathbb{R}$.

2. For each n-ary operation F on $\mathbb{R}$, let $^*F$ be the operation $f_R^{^*\mathfrak{R}}$ assigned to the symbol f_F by $^*\mathfrak{R}$. F is then the restriction to $\mathbb{R}$ of $^*F$.

Observe that $c_r^{^*\mathfrak{R}} = r$, so we need no special notation for this.

There is a general method (to be used heavily in the remainder of this section) for demonstrating properties of a relation $^*R$ or an operation $^*F$. One simply observes (1) that R or F has the property, (2) that the property can be expressed by a sentence of the language, and (3) that $\mathfrak{R} \equiv {}^*\mathfrak{R}$.

For example, the binary relation $^*<$ on $^*\mathbb{R}$ is transitive. This is because $<$ is transitive, and this property can be expressed by the sentence

$$\forall x \, \forall y \, \forall z (x P_< y \rightarrow y P_< z \rightarrow x P_< z).$$

By similar reasoning, $^*<$ satisfies trichotomy on $^*\mathbb{R}$ and thus is an ordering relation on $^*\mathbb{R}$.

For another example, we can prove that the binary operation $^*+$ on $^*\mathbb{R}$ is commutative, since $+$ is commutative and the commutative law can be expressed by a sentence. By applying this reasoning to each of the field axioms, we see that $(^*\mathbb{R}; 0, 1, {}^*+, {}^*\cdot)$ is a field.

This general method is used so much, we will shortly begin to take it for granted. If, for example, we assert that $^*|a \, ^*+ \, b| \, ^*\leq \, ^*|a| \, ^*+ \, ^*|b|$ for a and b in $^*\mathbb{R}$, we will take for granted that the reader perceives that the general method yields this fact.

We have $\mathbb{R} \subseteq {}^*\mathbb{R}$, but $\mathbb{R} \neq {}^*\mathbb{R}$. For we have some point b such that $\models_{^*\mathfrak{R}} c_r P_< v_1 [\![b]\!]$; i.e., $r \, ^*< \, b$. Thus b is infinitely large, being larger (in the ordering $^*<$) than any standard r, i.e., any $r \in \mathbb{R}$. Its reciprocal $1 \, ^*/ \, b$ will be a sample infinitesimal.

Properties of $\mathfrak{R}$ that *cannot* be expressed in the language are likely to fail in $^*\mathfrak{R}$. The least-upper-bound property is one such. There are non-empty bounded subsets S of $^*\mathbb{R}$ that have no least upper bound (with respect to the ordering $^*<$). For example, $\mathbb{R}$ is such a subset of $^*\mathbb{R}$. It is bounded by the infinite b of the preceding paragraph. But it has no least upper bound; see Exercise 7.

Define the set $\mathcal{F}$ of *finite* elements by the equation

$$\mathcal{F} = \{x \in {}^*\mathbb{R} \mid {}^*|x| \, ^*< \, y \quad \text{for some} \quad y \in \mathbb{R}\}.$$

Similarly, define the set $\mathcal{I}$ of *infinitesimals* by the equation

$$\mathcal{I} = \{x \in {}^*\mathbb{R} \mid {}^*|x| \; {}^*< y \quad \text{for all positive} \quad y \in \mathbb{R}\}.$$

If $A \subseteq \mathbb{R}$ is unbounded, then ${}^*A$ contains infinite points. For the sentence "for any real r there is an element $a \in A$ larger than r" is true and formalizable. Take some infinite positive b; there must be a larger (and hence infinite) member of ${}^*A$. For example, ${}^*\mathbb{N}$ contains infinite numbers.

The only standard infinitesimal, i.e., the only member of $\mathbb{R} \cap \mathcal{I}$, is 0. But there are other infinitesimals. For by the usual (formalizable) rules for inequalities, the reciprocal of any infinite number is an infinitesimal.

Algebraic Properties

In the next theorem we collect some algebraic facts about $\mathcal{F}$ and $\mathcal{I}$ that will be useful later.

THEOREM 28A   (a) $\mathcal{F}$ is closed under addition ${}^*+$, subtraction ${}^*-$, and multiplication ${}^*\cdot$.
(b) $\mathcal{I}$ is closed under addition ${}^*+$, subtraction ${}^*-$, and multiplication from $\mathcal{F}$:

$$x \in \mathcal{I} \quad \text{and} \quad z \in \mathcal{F} \Rightarrow x \; {}^*\cdot z \in \mathcal{I}.$$

In algebraic terminology, part (a) says that $\mathcal{F}$ is a subring of the field ${}^*\mathbb{R}$, and part (b) says that $\mathcal{I}$ is an ideal in the ring $\mathcal{F}$. We will see a little later what the quotient ring $\mathcal{F}/\mathcal{I}$ is.

PROOF.   (a) Let x and y be finite, so that ${}^*|x| \; {}^*< a$, ${}^*|y| \; {}^*< b$ for standard a and b in $\mathbb{R}$. Then

$$ {}^*|x \; {}^*\pm y| \; {}^*\leq {}^*|x| \; {}^*+ {}^*|y| \; {}^*< a + b \in \mathbb{R},$$

whence $x \; {}^*+ y$, $x \; {}^*- y$ are finite. Also

$$ {}^*|x \; {}^*\cdot y| \; {}^*< a \cdot b \in \mathbb{R},$$

whence $x \; {}^*\cdot y$ is finite, too.
(b) Let x and y be infinitesimals. Then for any positive standard a, ${}^*|x| \; {}^*< a/2$ and ${}^*|y| \; {}^*< a/2$. Hence

$$ {}^*|x \; {}^*\pm y| \; {}^*< a/2 + a/2 = a,$$

so that $x \; {}^*+ y$ and $x \; {}^*- y$ are infinitesimal. If z is finite, then ${}^*|z| \; {}^*< b$ for some standard b. Since x is infinitesimal, we have ${}^*|x| \; {}^*< a/b$, whence

$$ {}^*|x \; {}^*\cdot z| \; {}^*< (a/b)b = a.$$

Thus $x \; {}^*\cdot z$ is also infinitesimal.   ⊣

DEFINITION. x is *infinitely close* to y (written $x \simeq y$) iff $x \,{}^{*}\!- y$ is infinitesimal.

THEOREM 28B   (a) $\simeq$ is an equivalence relation on ${}^{*}\mathbb{R}$.
   (b) If $u \simeq v$ and $x \simeq y$, then $u \,{}^{*}\!+ x \simeq v \,{}^{*}\!+ y$ and ${}^{*}\!- u \simeq {}^{*}\!- v$.
   (c) If $u \simeq v$ and $x \simeq y$ and x, y, u, v are finite, then $u \,{}^{*}\!\cdot x \simeq v \,{}^{*}\!\cdot y$.

PROOF.   This is a consequence of part (b) of the preceding theorem ($\mathcal{I}$ is an ideal in $\mathcal{F}$).
   (a) $\simeq$ is reflexive since 0 is infinitesimal. $\simeq$ is symmetric since the negative (${}^{*}\!-$) of an infinitesimal is infinitesimal. Finally, suppose that $x \simeq y$ and $y \simeq z$. Then

$$x \,{}^{*}\!- z = (x \,{}^{*}\!- y) \,{}^{*}\!+ (y \,{}^{*}\!- z) \in \mathcal{I}$$

since $\mathcal{I}$ is closed under addition.
   (b) If $u \simeq v$ and $x \simeq y$, then

$$(u \,{}^{*}\!+ x) \,{}^{*}\!- (v \,{}^{*}\!+ y) = (u \,{}^{*}\!- v) \,{}^{*}\!+ (x \,{}^{*}\!- y) \in \mathcal{I}$$

since $\mathcal{I}$ is closed under addition. Also ${}^{*}\!- u \simeq {}^{*}\!- v$ since $\mathcal{I}$ is closed under negation.

$$(c) \quad (u \,{}^{*}\!\cdot x) \,{}^{*}\!- (v \,{}^{*}\!\cdot y) = (u \,{}^{*}\!\cdot x) \,{}^{*}\!- (u \,{}^{*}\!\cdot y) \,{}^{*}\!+ (u \,{}^{*}\!\cdot y) \,{}^{*}\!- (v \,{}^{*}\!\cdot y)$$
$$= u \,{}^{*}\!\cdot (x \,{}^{*}\!- y) \,{}^{*}\!+ (u \,{}^{*}\!- v) \,{}^{*}\!\cdot y \in \mathcal{I}$$

since $\mathcal{I}$ is closed under multiplication from $\mathcal{F}$. $\dashv$

For standard r and s, we have $r \simeq s$ iff $r = s$, as 0 is the only standard infinitesimal.

LEMMA 28C   If $x \not\simeq y$ and at least one is finite, then there is a standard q strictly between x and y.

PROOF.   We may suppose that $x \,{}^{*}\!< y$. In fact, we may further suppose that $0 \,{}^{*}\!< x \,{}^{*}\!< y$; the case $x \,{}^{*}\!< y \,{}^{*}\!\le 0$ is similar and the case $x \,{}^{*}\!< 0 \,{}^{*}\!< y$ is trivial. Since $x \not\simeq y$ there is a standard b such that $0 < b \,{}^{*}\!< y \,{}^{*}\!- x$. Since x is finite we have $x \,{}^{*}\!< mb$ for some positive integer m; take the least such m. Then $x \,{}^{*}\!< mb \,{}^{*}\!< y$. (By the leastness of m, $(m-1)b \,{}^{*}\!\le x$. So $mb \,{}^{*}\!\le x \,{}^{*}\!+ b \,{}^{*}\!< y$.)
                                                                              $\dashv$

THEOREM 28D   Every $x \in \mathcal{F}$ is infinitely close to a unique $r \in \mathbb{R}$.

PROOF.   For $x \in \mathcal{F}$ the set

$$S = \{y \in \mathbb{R} \mid y \,{}^{*}\!< x\}$$

of standard points below x has an upper bound in $\mathbb{R}$. Let r be its least upper bound; we claim that $x \simeq r$.

If $x \not\simeq r$, then by the lemma there is a standard q between x and r. If $r < q \;^*\!< x$, then r fails to be an upper bound for S. If $x \;^*\!< q < r$, then q is also an upper bound for S, contradicting the leastness of r. Hence $x \simeq r$.

This establishes the existence of r. As for the uniqueness, note that if $x \simeq r$ and $x \simeq s$, then $r \simeq s$. For standard r and s this implies that $r = s$.   ⊣

COROLLARY 28E   Each finite x has a unique decomposition of the form $x = s \;^*\!+ i$, where s is standard and i is infinitesimal.

We call s the *standard part* of x, written st(x). (Another notation for the standard part of x is $^{\circ}x$.) Of course for standard r, st$(r) = r$. The next theorem summarizes some properties of the st function.

THEOREM 28F   (a) st maps $\mathcal{F}$ onto $\mathbb{R}$.
(b) st$(x) = 0$ iff x is infinitesimal.
(c) st$(x \;^*\!+ y) = \text{st}(x) + \text{st}(y)$.
(d) st$(x \;^*\!\cdot y) = \text{st}(x) \cdot \text{st}(y)$.

PROOF.   (a) and (b) are clear. Since st$(x) \simeq x$ and st$(y) \simeq y$, we have by part (b) of Theorem 28B that st$(x) + \text{st}(y) \simeq x \;^*\!+ y$. Hence the left side equals st$(x \;^*\!+ y)$. Part (d) is similar and uses part (c) of Theorem 28B.   ⊣

(In algebraic terminology, this theorem asserts that st is a homomorphism of the ring $\mathcal{F}$ onto the field $\mathbb{R}$, with kernel $\mathcal{I}$. Consequently, the quotient ring $\mathcal{F}/\mathcal{I}$ is isomorphic to the real field $\mathbb{R}$.)

Henceforth in this section we will streamline our notation by omitting the asterisks on the symbols for the arithmetic operations $^*\!+$, $^*\!-$, $^*\!\cdot$, and $^*\!/$.

Convergence

In calculus courses convergence is usually treated in terms of ε's and δ's and variables that come delicately close to certain values. We will give here the beginning of an alternative treatment of convergence, where variables come infinitely close to the limiting values.

DEFINITION.   Let $F : \mathbb{R} \to \mathbb{R}$. Then F *converges at a to b* iff whenever x is infinitely close to (but different from) a, then $^*F(x)$ is infinitely close to b.

PROOF OF EQUIVALENCE WITH THE ORDINARY DEFINITION.   First suppose that F converges at a to b in the ordinary sense. That is, for any $\varepsilon > 0$ there is a $\delta > 0$ such that

$$0 \neq |x - a| < \delta \;\Rightarrow\; |b - F(x)| < \varepsilon \quad \text{for any } x.$$

The displayed sentence (concerning the standard numbers ε and

δ) is formalizable and thus holds in $^*\mathfrak{R}$. Now if x in $^*\mathbb{R}$ is infinitely close to (but different from) a, then certainly $0 \neq {}^*|x - a| {}^*< \delta$. Hence $^*|b - {}^*F(x)| {}^*< \varepsilon$. Since ε was arbitrary, $b \simeq {}^*F(x)$.

Conversely, assume that the condition stated in the definition is met. Then for any standard $\varepsilon > 0$, the sentence

There exists $\delta > 0$ such that for all x, $0 \neq |a - x| < \delta \Rightarrow |b - F(x)| < \varepsilon$

(when formalized) holds in $^*\mathfrak{R}$, since we can take δ to be infinitesimal. Hence the sentence holds in $\mathfrak{R}$ also. ⊣

First remark: It is entirely possible that F does not converge at a to any number. On the other hand, F converges at a to at most one b. For if i is a nonzero infinitesimal, then $b = \text{st}(^*F(a + i))$. It is traditional to denote this b by "$\lim_{x \to a} F(x)$." Thus

$$\lim_{x \to a} F(x) = \text{st}(^*F(a + i)).$$

Second remark: It is not really necessary to have dom $F = \mathbb{R}$. It is enough for a to be an accumulation point of dom F. (a is an accumulation point of S iff a is infinitely close to, but different from, some member of $^*S$.)

COROLLARY 28G F is continuous at a iff whenever $x \simeq a$, then $^*F(x) \simeq F(a)$.

Now consider a function $F : \mathbb{R} \to \mathbb{R}$ and a standard $a \in \mathbb{R}$. Then the derivative $F'(a)$ is

$$\lim_{h \to 0} \frac{F(a + h) - F(a)}{h}$$

By our definition of limit, this can also be stated: $F'(a) = b$ iff for every nonzero infinitesimal dx we have $dF/dx \simeq b$, where $dF = {}^*F(a + dx) - F(a)$. Thus if there is such a b (i.e., if $F'(a)$ exists), then

$$F'(a) = \text{st}(dF/dx)$$

for any nonzero infinitesimal dx. Here dF/dx is the result of *dividing* dF by dx. The fact that we simply use division here greatly facilitates calculations.

EXAMPLE. Let $F(x) = x^2$. Then $F'(a) = 2a$, since

$$\frac{dF}{dx} = \frac{(a + dx)^2 - a^2}{dx} = \frac{2a(dx) + (dx)^2}{dx} = 2a + dx \simeq 2a.$$

THEOREM 28H If $F'(a)$ exists, then F is continuous at a.

PROOF. For any nonzero infinitesimal dx we have

$$\frac{^*F(a + dx) - F(a)}{dx} \simeq F'(a).$$

The right side is standard, so the left side is at least finite. Consequently, when we multiply the left side by the infinitesimal dx we are left with the fact that $^*F(a + dx) - F(a) \in \mathcal{I}$; i.e., $^*F(a + dx) \simeq F(a)$. ⊣

The reader should note that this result is not some nonstandard analogue of a classical theorem, nor even a generalization of a classical theorem. It *is* a classical theorem. It is only the proof that is nonstandard. The same remarks apply to the next theorem. Let $F \circ G$ be the function whose value at a is $F(G(a))$.

CHAIN RULE   Assume that $G'(a)$ and $F'(G(a))$ exist. Then $(F \circ G)'(a)$ exists and equals $F'(G(a)) \cdot G'(a)$.

PROOF.   First, notice that $^*(F \circ G) =^* F \circ {}^*G$, since the sentence $\forall v_1 f_{F \circ G} v_1 = f_F f_G v_1$ holds in the structures. Now consider any nonzero infinitesimal dx. Let

$$
\begin{aligned}
dG &= {}^*G(a + dx) - G(a), \\
dF &= {}^*(F \circ G)(a + dx) - (F \circ G)(a) \\
&= {}^*F({}^*G(a + dx)) - F(G(a)) \\
&= {}^*F(G(a) + dG) - F(G(a)).
\end{aligned}
$$

Then $dG \simeq 0$ since G is continuous at a. If $dG \neq 0$, then by the last of these equations $dF/dG \simeq F'(G(a))$, whence

$$
\frac{dF}{dx} = \frac{dF}{dG} \cdot \frac{dG}{dx} \simeq F'(G(a)) \cdot G'(a).
$$

If $dG = 0$, then $dF = 0$ and $G'(a) \simeq dG/dx = 0$, so we again have

$$
\frac{dF}{dx} \simeq F'(G(a)) \cdot G'(a).
$$

⊣

These theorems are but samples of the treatment of convergence in terms of infinite proximity. The method is not at all limited to elementary topics. One can construct delta functions δ with the property that $\int_{-\infty}^{\infty} \delta = 1$ and yet $\delta(x) \simeq 0$ for $x \not\simeq 0$. Original results in analysis (e.g., in the theory of Hilbert spaces) have been obtained by the method of nonstandard analysis. Possibly the method will be more widely used in the future as more analysts become familiar with it.

Exercises

1. ($\mathbb{Q}$ is dense in $\mathbb{R}$.) Let $\mathbb{Q}$ be the set of rational numbers. Show that every member of $^*\mathbb{R}$ is infinitely close to some member of $^*\mathbb{Q}$.

2. (a) Let $A \subseteq \mathbb{R}$ and $F : A \to \mathbb{R}$. Then F is also a binary relation on $\mathbb{R}$; show that $^*F : {}^*A \to {}^*\mathbb{R}$.
   (b) Let $S : \mathbb{N} \to \mathbb{R}$. Recall that S is said to converge to b iff for every $\varepsilon > 0$ there is some k such that for all $n > k$, $|S(n) - b| < \varepsilon$.

Show that this is equivalent to the condition: $^*S(x) \simeq b$ for every infinite $x \in {}^*\mathbb{N}$.

(c) Assume that $S_i : \mathbb{N} \to \mathbb{R}$ and S_i converges to b_i for $i = 1, 2$. Show that $S_1 + S_2$ converges to $b_1 + b_2$ and $S_1 \cdot S_2$ converges to $b_1 \cdot b_2$.

3. Let $F : A \to \mathbb{R}$ be one-to-one, where $A \subseteq \mathbb{R}$. Show that if $x \in {}^*A$ but $x \notin A$, then $^*F(x) \notin \mathbb{R}$.

4. Let $A \subseteq \mathbb{R}$. Show that $A = {}^*A$ iff A is finite.

5. (Bolzano–Weierstrass theorem) Let $A \subseteq \mathbb{R}$ be bounded and infinite. Show that there is a point $p \in \mathbb{R}$ that is infinitely close to, but different from, some member of $^*A$. *Suggestion*: Let $S : \mathbb{N} \to A$ with S one-to-one; look at $^*S(x)$ for infinite $x \in {}^*\mathbb{N}$.

6. (a) Show that $^*\mathbb{Q}$ has cardinality at least $2^{\aleph_0}$, where $\mathbb{Q}$ is the set of rational numbers. Suggestion: Use Exercise 1.

(b) Show that $^*\mathbb{N}$ has cardinality at least $2^{\aleph_0}$.

7. Let A be a subset of $\mathbb{R}$ having no greatest member. Then as a subset of $^*\mathbb{R}$, A will have upper bounds (with respect to the ordering $^*<$) in $^*\mathbb{R}$. But show that A does not have a least such bound.

Chapter THREE

Undecidability

SECTION 3.0
Number Theory

In this chapter we will focus our attention on a specific language, the language of number theory. This will be the first-order language with equality and with the following parameters:

$\forall$, intended to mean "for all natural numbers." (Recall that the set $\mathbb{N}$ of natural numbers is the set $\{0, 1, 2, \ldots\}$. Zero is natural.)

$\mathbf{0}$, a constant symbol intended to denote the number 0.

$\mathbf{S}$, a one-place function symbol intended to denote the successor function $S : \mathbb{N} \to \mathbb{N}$, i.e., the function for which $S(n) = n + 1$.

$<$, a two-place predicate symbol intended to denote the usual (strict) ordering relation on $\mathbb{N}$.

$+, \cdot, \mathbf{E}$, two-place function symbols intended to denote the operations $+$, $\cdot$, and E of addition, multiplication, and exponentiation, respectively.

We will let $\mathfrak{N}$ be the intended structure for this language. Thus we may informally write

$$\mathfrak{N} = (\mathbb{N}; 0, S, <, +, \cdot, E).$$

(More precisely, $|\mathfrak{N}| = \mathbb{N}$, $\mathbf{0}^{\mathfrak{N}} = 0$, and so forth.)

By *number theory* we mean the theory of this structure, Th $\mathfrak{N}$. As warmup exercises we will study (in Sections 3.1 and 3.2) certain reducts of $\mathfrak{N}$,

i.e., restrictions of $\mathfrak{N}$ to sublanguages:

$$\mathfrak{N}_S = (\mathbb{N}; 0, S),$$
$$\mathfrak{N}_L = (\mathbb{N}; 0, S, <),$$
$$\mathfrak{N}_A = (\mathbb{N}; 0, S, <, +).$$

Finally, in Section 3.8 we will consider

$$\mathfrak{N}_M = (\mathbb{N}; 0, S, <, +, \cdot).$$

For each of these structures we will raise the same questions:

(A) Is the theory of the structure decidable? If so, what is a nice set of axioms for the theory? Is there a finite set of axioms?

(B) What subsets of $\mathbb{N}$ are definable in the structure?

(C) What do the nonstandard models of the theory of the structure look like? (By "nonstandard" we mean "not isomorphic to the intended structure.")

Our reason for choosing number theory (rather than, say, group theory) for special study is this: We can show that a certain subtheory of number theory is an undecidable set of sentences. We will also be able to infer that any satisfiable theory that is at least as strong as this fragment of number theory (e.g., the full number theory or set theory) must be undecidable. In particular, such a theory cannot be both complete and axiomatizable.

In order to show that our subtheory of number theory is undecidable, we will show that it is strong enough to represent (in a sense to be made precise) facts about sequences of numbers, certain operations on numbers, and ultimately facts about decision procedures. This last feature then lets us perform a diagonal argument that demonstrates undecidability.

We could alternatively use, in place of a subtheory of number theory, some other theory (such as a fragment of the theory of finite sets) in which we could conveniently represent facts about decision procedures.

Before giving examples of the expressiveness of the language of number theory, it is convenient to introduce some notational conventions. As a concession to everyday usage, we will write

$$x < y, \quad x + y, \quad x \cdot y, \quad \text{and} \quad x \, \mathbf{E} \, y$$

in place of the official

$$< xy, \quad + xy, \quad \cdot xy, \quad \text{and} \quad \mathbf{E} xy.$$

For each natural number k we have a term $\mathbf{S}^k \mathbf{0}$ (the *numeral* for k) that denotes it:

$$\mathbf{S}^0 \mathbf{0} = \mathbf{0}, \quad \mathbf{S}^1 \mathbf{0} = \mathbf{S0}, \quad \mathbf{S}^2 \mathbf{0} = \mathbf{SS0}, \quad \text{etc.}$$

(The set of numerals is generated from $\{\mathbf{0}\}$ by the operation of

prefixing **S**.) The fact that every natural number can be named in the language will be a useful feature.

Even though only countably many relations on $\mathbb{N}$ are definable in $\mathfrak{N}$, almost all the familiar relations are definable. For example, the set of primes is defined in $\mathfrak{N}$ by

$$v_1 \neq \mathbf{S}^1\mathbf{0} \wedge \forall\, v_2 \,\forall\, v_3(v_1 = v_2 \cdot v_3 \rightarrow v_2 = \mathbf{S}^1\mathbf{0} \vee v_3 = \mathbf{S}^1\mathbf{0}).$$

Later we will find it important to show that many other specific relations are definable in $\mathfrak{N}$.

One naturally expects the expressiveness of the language to be severely restricted when some of the parameters are omitted. For example, the set of primes, as we shall see, is not definable in $\mathfrak{N}_A$. On the other hand, in Section 3.8 we will show that any relation definable in $\mathfrak{N}$ is also definable in $\mathfrak{N}_M$.

Preview

The main theorems of this chapter — the theorems associated with the names of Gödel, Tarski, and Church — are proved in Section 3.5. But we can already sketch here some of the ideas involved. We want to compare the concepts of *truth* and *proof*; that is, we want to compare the set of sentences *true* in $\mathfrak{N}$ with the set of sentences that might be *provable* from an appropriate set A of axioms.

We can assign to each formula α of the language of number theory an integer $\sharp\alpha$, called the Gödel number of α. Any sufficiently straightforward way of assigning distinct integers to formulas would suffice for our purposes; a particular assignment is adopted at the beginning of Section 3.4. What *is* important is that from α we can effectively find the number $\sharp\alpha$, and conversely. Similarly, to each finite sequence D of formulas (such as a deduction) we assign an integer $\mathcal{G}(D)$. Note that for any *set* A of formulas, we can form the corresponding set $\{\sharp\alpha \mid \alpha \in A\}$ of numbers.

There are now three ways in which to proceed: the *self-reference approach*, the *diagonalization approach*, and the *computability approach*. It will be argued later, however, that the three approaches are more closely related than they appear — they are three aspects of one approach.

First, in the *self-reference* approach, we make a sentence σ that can be thought of as saying, "I am unprovable." More specifically, we have the following:

> **THEOREM 30A**   Let $A \subseteq \mathrm{Th}\,\mathfrak{N}$ be a set of sentences true in $\mathfrak{N}$, and assume that the set $\{\sharp\alpha \mid \alpha \in A\}$ of Gödel numbers of members of A is a set definable in $\mathfrak{N}$. Then we can find a sentence σ such that σ is true in $\mathfrak{N}$ but σ is not deducible from A.

Proof.   We will construct σ to express (in an indirect way) that σ itself is not a theorem of A. Then the argument will go roughly as follows: If $A \vdash \sigma$, then what σ says is false, contradicting the fact that A consists of true sentences. And so $A \nvdash \sigma$, whence σ is true.

To construct σ, we begin by considering the ternary relation R defined by

$\langle a, b, c \rangle \in R$ iff a is the Gödel number of some formula α and c is the value of $\mathcal{G}$ at some deduction from A of $\alpha(\mathbf{S}^b\mathbf{0})$.

Then because $\{\sharp \alpha \mid \alpha \in A\}$ is definable in $\mathfrak{N}$, it follows that R is definable also. (The details of this step must wait until later sections.) Let ρ be a formula that defines R in $\mathfrak{N}$. Let q be the Gödel number of

$$\forall\, v_3 \,\neg\, \rho(v_1, v_1, v_3).$$

(We use here the notation: $\varphi(t) = \varphi_t^{v_1}$, $\varphi(t_1, t_2) = (\varphi_{t_1}^{v_1})_{t_2}^{v_2}$, and so forth.) Then let σ be

$$\forall\, v_3 \,\neg\, \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3).$$

Thus σ says that no number is the value of $\mathcal{G}$ at a deduction from A of the result of replacing, in formula number q, the variable v_1 by the numeral for q; i.e., no number is the value of $\mathcal{G}$ at a deduction of σ.

Suppose that, contrary to our expectations, there is a deduction of σ from A. Let k be the value of $\mathcal{G}$ at a deduction. Then $\langle q, q, k \rangle \in R$ and hence

$$\models_{\mathfrak{N}} \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^k\mathbf{0}).$$

It is clear that

$$\sigma \vdash \neg\, \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^k\mathbf{0})$$

and the two displayed lines tell us that σ is false in $\mathfrak{N}$. But $A \vdash \sigma$ and the members of A are true in $\mathfrak{N}$, so we have a contradiction.

Hence there is no deduction of σ from A. And so for every k, we have $\langle q, q, k \rangle \notin R$. Thus for every k

$$\models_{\mathfrak{N}} \neg\, \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^k\mathbf{0}),$$

from which it follows (with the help of the substitution lemma) that

$$\models_{\mathfrak{N}} \forall\, v_3 \,\neg\, \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3);$$

i.e., σ is true in $\mathfrak{N}$. $\dashv$

We will argue later — using something called Church's thesis — that any decidable set of natural numbers must be definable in $\mathfrak{N}$. The conclusion will then be that Th $\mathfrak{N}$ is not axiomatizable.

COROLLARY 30B   The set $\{\sharp\tau \mid \models_\mathfrak{N} \tau\}$ of Gödel numbers of sentences true in $\mathfrak{N}$ is a set that is not definable in $\mathfrak{N}$.

PROOF.   If this set were definable, we could take $A = \mathrm{Th}\,\mathfrak{N}$ in the preceding theorem to obtain a contradiction.   ⊣

Section 3.5 will follow the self-reference approach, but with a variation in which the sentence σ tries to say, "I am false." (The well-known liar paradox is relevant here!)

But if this "self-reference" construction seems too much like a magic trick, there is a second way to describe the situation: the *diagonalization approach*, which does not use an obvious self-reference.

We start by defining the following binary relation P on the natural numbers:

$$\langle a, b\rangle \in P \iff \begin{array}{l} a \text{ is the Gödel number of a formula } \alpha(v_1) \\ \text{(with just } v_1 \text{ free) and } \models_\mathfrak{N} \alpha(\mathbf{S}^b\mathbf{0}). \end{array}$$

(More informally, $\langle a, b\rangle \in P \Leftrightarrow$ "a is true of b.") Then any set of natural numbers that is definable in $\mathfrak{N}$ equals, for some a, the "vertical section"

$$P_a = \{b \mid \langle a, b\rangle \in P\}$$

of P. Namely, we take a to be the Gödel number of a formula defining the set, and use the fact that $\models_\mathfrak{N} \alpha(\mathbf{S}^b\mathbf{0}) \Leftrightarrow \models_\mathfrak{N} \alpha(v_1)[\![b]\!]$.

So any definable (in $\mathfrak{N}$) set of natural numbers is somewhere on the list $P_1, P_2, \ldots$. Now we "diagonalize out" of the list. Define the set:

$$H = \{b \mid \langle b, b\rangle \notin P\}.$$

(More informally, $b \in H \Leftrightarrow$ "b is not true of b.") Then H is nowhere on the list $P_1, P_2, \ldots$. ($H \neq P_3$ because $3 \in H \Leftrightarrow 3 \notin P_3$, so the number 3 belongs to exactly one of these two sets and not to the other.) Therefore H is not definable in $\mathfrak{N}$.

Why is H undefinable? After all, we have above specified that

$$b \in H \iff \begin{array}{l} \text{not } [b \text{ is not the Gödel number of a formula } \alpha(v_1) \\ \text{(with just } v_1 \text{ free) and } \models_\mathfrak{N} \alpha(\mathbf{S}^b\mathbf{0})]. \end{array}$$

What is the barrier to translating this specification into the language of arithmetic? We will show that the barrier is *not* being the Gödel number of a formula — we can translate that — and the barrier is not having v_1 free and not substituting the numeral $\mathbf{S}^b\mathbf{0}$ into a formula. By the process of elimination, we will show that the only possible barrier is saying of a sentence that is true in $\mathfrak{N}$.

THEOREM 30C   (a) The set $\{\sharp\tau \mid \models_\mathfrak{N}\tau\}$ of Gödel numbers of sentences true in $\mathfrak{N}$ is not definable in $\mathfrak{N}$.
    ⋆(b) The theory $\mathrm{Th}\,\mathfrak{N}$ is undecidable.
    ⋆(c) The theory $\mathrm{Th}\,\mathfrak{N}$ is not axiomatizable.

PROOF.   Part (a), which is the same as Corollary 30B in the self-reference approach, has the diagonal proof sketched above. That is, if to the contrary Th $\mathfrak{N}$ were definable in $\mathfrak{N}$, then the above set H would also be definable, which it is not.

Part (b) will then follow, after we argue every decidable set of natural numbers must be definable in $\mathfrak{N}$. If Th $\mathfrak{N}$ were decidable, then the corresponding set $\{\sharp\tau \mid \models_{\mathfrak{N}} \tau\}$ of numbers would be decidable and hence definable in $\mathfrak{N}$, which it is not.

And part (c) is an immediate consequence of part (b) and Corollary 26I, since Th $\mathfrak{N}$ is a complete theory.   ⊣

And thirdly, the *computability approach* presents us with a stark difference between what is true and what is provable. From Section 2.6 we know that whenever A is a decidable set (or even an effectively enumerable set) of axioms we might choose for Th $\mathfrak{N}$, the set Cn A of provable sentence will be an effectively enumerable set.

In contrast, the computability approach will show — using Church's thesis — that the set Th $\mathfrak{N}$ of all true sentences is *not* effectively enumerable. This fact, which is closely related to Theorem 30C, will follow from another diagonal argument in Section 3.6.

*THEOREM 30D   For any decidable (or even effectively enumerable) set A of axioms,

$$\text{Cn } A \neq \text{Th } \mathfrak{N}$$

because the set on the left is effectively enumerable and the set on the right is not.

Theorem 30D presents the dilemma: Either the axioms are lying to us by allowing us to deduce false sentences, or else the axioms are incomplete, in the sense that some true sentences cannot be deduced from those axioms.

This computability approach is implicit in parts of Section 3.5, but it is in Section 3.6 that the approach explicitly appears, and where it is compared with the other two approaches.

SECTION 3.1
Natural Numbers with Successor

We begin with a situation that is simple enough to let us give reasonably complete answers to our questions. We reduce the set of parameters to just $\forall$, $\mathbf{0}$, and $\mathbf{S}$, eliminating $<$, $+$, $\cdot$, and $\mathbf{E}$. The corresponding reduct of $\mathfrak{N}$ is

$$\mathfrak{N}_S = (\mathbb{N}; 0, S).$$

In this restricted language we still have the numerals, naming each point in $\mathbb{N}$. But the sentences we can express in the language are, from the viewpoint of arithmetic, uninteresting.

We want to ask about $\mathfrak{N}_S$ the same questions that interest us in the case of $\mathfrak{N}$. We want to know about the complexity of the set Th $\mathfrak{N}_S$; we want to study definability in $\mathfrak{N}_S$; and we want to survey the nonstandard models of $\mathfrak{N}_S$.

To study the theory of the natural numbers with successor (Th $\mathfrak{N}_S$), we begin by listing a few of its members, i.e., sentences true in $\mathfrak{N}_S$. (These sentences will ultimately provide an axiomatization for the theory.)

S1. $\forall x \, \mathbf{S}x \neq \mathbf{0}$, a sentence asserting that zero has no predecessor.

S2. $\forall x \, \forall y (\mathbf{S}x = \mathbf{S}y \rightarrow x = y)$. This asserts that the successor function is one-to-one.

S3. $\forall y (y \neq \mathbf{0} \rightarrow \exists x \, y = \mathbf{S}x)$. This asserts that any nonzero number is the successor of something.

S4.1 $\forall x \, \mathbf{S}x \neq x$.

S4.2 $\forall x \, \mathbf{SS}x \neq x$.

$\ldots$

S4.n $\forall x \, \mathbf{S}^n x \neq x$, where the superscript n indicates that the symbol $\mathbf{S}$ occurs at n consecutive places.

Let A_S be the set consisting of the above sentences S1, S2, S3, S4.n ($n = 1, 2, \ldots$). Clearly these sentences are true in $\mathfrak{N}_S$; i.e., $\mathfrak{N}_S$ is a model of A_S. Hence

$$\operatorname{Cn} A_S \subseteq \operatorname{Th} \mathfrak{N}_S.$$

(Anything true in every model of A_S is true in this model.) What is not so obvious is that equality holds. We will prove this by considering arbitrary models of A_S.

What can be said of an arbitrary model

$$\mathfrak{A} = (|\mathfrak{A}|; \mathbf{0}^{\mathfrak{A}}, \mathbf{S}^{\mathfrak{A}})$$

of the axioms A_S? $\mathbf{S}^{\mathfrak{A}}$ must be a one-to-one map of $|\mathfrak{A}|$ onto $|\mathfrak{A}| - \{\mathbf{0}^{\mathfrak{A}}\}$, by S1, S2, and S3. And by S4.n, there can be no loops of size n. Thus $|\mathfrak{A}|$ must contain the "standard" points:

$$\mathbf{0}^{\mathfrak{A}} \rightarrow \mathbf{S}^{\mathfrak{A}}(\mathbf{0}^{\mathfrak{A}}) \rightarrow \mathbf{S}^{\mathfrak{A}}(\mathbf{S}^{\mathfrak{A}}(\mathbf{0}^{\mathfrak{A}})) \rightarrow \ldots,$$

which are all distinct. The arrow here indicates the action of $\mathbf{S}^{\mathfrak{A}}$. There may or may not be other points. If there is another point a in $|\mathfrak{A}|$, then there will be the successor of a, its successor, etc. Not only that, but since (by S3) each nonzero element has a predecessor (something of which it is the successor) which is (by S2) unique, $|\mathfrak{A}|$ must contain the predecessor of a, its predecessor, etc. These must all be distinct lest

there be a finite loop. Thus a belongs to a "Z-chain":

$$\cdots * \to * \to a \to \mathbf{S}^{\mathfrak{A}}(a) \to \mathbf{S}^{\mathfrak{A}}(\mathbf{S}^{\mathfrak{A}}(a)) \to \cdots.$$

(We refer to these as Z-chains because they are arranged like the set $\mathbb{Z}$ of all integers $\{\ldots, -1, 0, 1, 2, \ldots\}$.) There can be any number of Z-chains. But any two Z-chains must be disjoint, as S2 prohibits merging. Similarly, any Z-chain must be disjoint from the standard part.

This can be restated in another way. Say that two points a and b in $|\mathfrak{A}|$ are *equivalent* if the function $\mathbf{S}^{\mathfrak{A}}$ can be applied a finite number of times to one point to yield the other point. This *is* an equivalence relation. (It is clearly reflexive and symmetric; the transitivity follows from the fact that $\mathbf{S}^{\mathfrak{A}}$ is one-to-one.) The standard part of $|\mathfrak{A}|$ is the equivalence class containing $\mathbf{0}^{\mathfrak{A}}$. For any other point (if any) a in $|\mathfrak{A}|$, the equivalence class of a is the set generated from $\{a\}$ by $\mathbf{S}^{\mathfrak{A}}$ and its inverse. This equivalence class is the Z-chain described above.

Conversely, any structure $\mathfrak{B}$ (for this language) that has a standard part

$$\mathbf{0}^{\mathfrak{B}} \to \mathbf{S}^{\mathfrak{B}}(\mathbf{0}^{\mathfrak{B}}) \to \mathbf{S}^{\mathfrak{B}}(\mathbf{S}^{\mathfrak{B}}(\mathbf{0}^{\mathfrak{B}})) \to \cdots$$

and a nonstandard part consisting of any number of separate Z-chains is a model of A_S. (Check through the list of axioms in A_S, and note that each is true in $\mathfrak{B}$.) We thus have a complete characterization of what the models of A_S must look like.

If a model $\mathfrak{A}$ of A_S has only countably many Z-chains, then $|\mathfrak{A}|$ is countable. In general, if the set of Z-chains has cardinality[1] λ, then altogether the number of points in $|\mathfrak{A}|$ is $\aleph_0 + \aleph_0 \cdot \lambda$. By facts of cardinal arithmetic (cf. Chapter 0) this number is the larger of $\aleph_0$ and λ. Hence

$$\text{card } |\mathfrak{A}| = \begin{cases} \aleph_0 & \text{if } \mathfrak{A} \text{ has countably many Z-chains,} \\ \lambda & \text{if } \mathfrak{A} \text{ has an uncountable number } \lambda \text{ of Z-chains.} \end{cases}$$

LEMMA 31A   If $\mathfrak{A}$ and $\mathfrak{A}'$ are models of A_S having the same number of Z-chains, then they are isomorphic.

PROOF.   There is a unique isomorphism between the standard part of $\mathfrak{A}$ and the standard part of $\mathfrak{A}'$. By hypothesis we are given a one-to-one correspondence between the set of Z-chains of $\mathfrak{A}$ and the set of Z-chains of $\mathfrak{A}'$; thus each chain of $\mathfrak{A}$ is paired with a chain of $\mathfrak{A}'$. Clearly any two Z-chains are isomorphic. By combining all the individual isomorphisms (which uses the axiom of choice) we have an isomorphism of $\mathfrak{A}$ onto $\mathfrak{A}'$.   ⊣

Thus a model of A_S is determined to within isomorphism by its number of Z-chains. For $\mathfrak{N}_S$ this number is zero, but any number is possible.

---

[1] To avoid uncountable cardinals, see Exercise 3.

The reader should note that there is no sentence of the language which says, "There are no Z-chains." In fact, there is no set Σ of sentences such that a model $\mathfrak{A}$ of A_S satisfies Σ iff $\mathfrak{A}$ has no Z-chains. For by the LST theorem there is an uncountable structure $\mathfrak{A}$ with $\mathfrak{A} \equiv \mathfrak{N}_S$. But $\mathfrak{A}$ has uncountably many Z-chains and $\mathfrak{N}_S$ has none.

THEOREM 31B   Let $\mathfrak{A}$ and $\mathfrak{B}$ be uncountable models of A_S of the same cardinality. Then $\mathfrak{A}$ is isomorphic to $\mathfrak{B}$.

PROOF.   By the above discussion, $\mathfrak{A}$ has card $\mathfrak{A}$ Z-chains, and $\mathfrak{B}$ has card $\mathfrak{B}$ Z-chains. Since card $\mathfrak{A}$ = card $\mathfrak{B}$, they have the same number of Z-chains and hence are isomorphic.   ⊣

THEOREM 31C   Cn A_S is a complete theory.

PROOF.   Apply the Łoś–Vaught test of Section 2.6. The preceding theorem asserts that the theory Cn A_S is categorical in any un-countable power. Furthermore, A_S has no finite models. Hence the Łoś–Vaught test applies.   ⊣

COROLLARY 31D   Cn A_S = Th $\mathfrak{N}_S$.

PROOF.   We have Cn $A_S \subseteq$ Th $\mathfrak{N}_S$; the first theory is complete and the second is satisfiable.   ⊣

★COROLLARY 31E   Th $\mathfrak{N}_S$ is decidable.

PROOF.   Any complete and axiomatizable theory is decidable (by Corollary 25G). A_S is a decidable set of axioms for this theory.
                                                                    ⊣

Elimination of Quantifiers

Once one knows a theory to be decidable, it is tempting to try to find a realistically practical decision procedure. We will give such a procedure for Th $\mathfrak{N}_S$, based on "elimination of quantifiers."

DEFINITION.   A theory T admits elimination of quantifiers iff for every formula φ there is a quantifier-free formula ψ such that

$$T \models (\varphi \leftrightarrow \psi).$$

Actually it is enough to consider only formulas ψ of a rather special form:

THEOREM 31F   Assume that for every formula φ of the form

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_n),$$

where each α_i is an atomic formula or the negation of an atomic formula, there is a quantifier-free formula ψ such that $T \models (\varphi \leftrightarrow \psi)$. Then T admits elimination of quantifiers.

PROOF.   First we claim that we can find a quantifier-free equivalent
for any formula of the form $\exists\, x\, \theta$ for quantifier-free θ. We begin
by putting θ into disjunctive normal form (Corollary 15C). The
resulting formula,

$$\exists\, x[(\alpha_0 \wedge \cdots \wedge \alpha_m) \vee (\beta_0 \wedge \cdots \wedge \beta_n) \vee \cdots \vee (\xi_0 \wedge \cdots \wedge \xi_t)],$$

is logically equivalent to

$$\exists\, x(\alpha_0 \wedge \cdots \wedge \alpha_m) \vee \exists\, x(\beta_0 \wedge \cdots \wedge \beta_n) \vee \cdots$$
$$\vee \exists\, x(\xi_0 \wedge \cdots \wedge \xi_t).$$

By assumption, each disjunct of this formula can be replaced by
a quantifier-free formula.

   We leave it to the reader to show (in Exercise 2) that by using
the above paragraph one can obtain a quantifier-free equivalent
for an arbitrary formula.   ⊣

   In the special case where the theory in question is the theory $\mathrm{Th}\,\mathfrak{A}$ of
a structure $\mathfrak{A}$, the definition can be restated: $\mathrm{Th}\,\mathfrak{A}$ admits elimination of
quantifiers iff for every formula φ there is a quantifier-free formula ψ
such that φ and ψ are "equivalent in $\mathfrak{A}$"; i.e.,

$$\models_{\mathfrak{A}} (\varphi \leftrightarrow \psi)[s]$$

for any map s of the variables into $|\mathfrak{A}|$.

THEOREM 31G $\mathrm{Th}\,\mathfrak{N}_S$ admits elimination of quantifiers.

PROOF.   By the preceding theorem, it suffices to consider a formula

$$\exists\, x(\alpha_0 \wedge \cdots \wedge \alpha_q),$$

where each α_i is atomic or is the negation of an atomic formula.
We will describe how to replace this formula by another that is
quantifier-free. The equivalence of the new formula to the given
one will, in fact, be a consequence of A_S; see Exercise 3.

   In the language of $\mathfrak{N}_S$ the only terms are of the form $\mathbf{S}^k u$, where
u is $\mathbf{0}$ or a variable. The only atomic formulas are equations. We
may suppose that the variable x occurs in each α_i. For if x does
not occur in α, then

$$\exists\, x(\alpha \wedge \beta) \models\dashv \alpha \wedge \exists\, x\, \beta.$$

Thus each α_i has the form

$$\mathbf{S}^m x = \mathbf{S}^n u$$

or the negation of this equation, where u is $\mathbf{0}$ or a variable. We
may further suppose u is different from x, since $\mathbf{S}^m x = \mathbf{S}^n x$ could
be replaced by $\mathbf{0} = \mathbf{0}$ if $m = n$, and by $\mathbf{0} \neq \mathbf{0}$ if $m \neq n$.

Case 1: Each α_i is the negation of an equation. Then the formula may be replaced by $\mathbf{0} = \mathbf{0}$. (Why?)

Case 2: There is at least one α_i not negated; say α_0 is

$$\mathbf{S}^m x = t,$$

where the term t does not contain x. Since the solution for x must be non-negative, we replace α_0 by

$$t \neq \mathbf{0} \wedge \cdots \wedge t \neq \mathbf{S}^{m-1}\mathbf{0}$$

(or by $\mathbf{0} = \mathbf{0}$ if $m = 0$). Then in each other α_j we replace, say,

$$\mathbf{S}^k x = u$$

first by

$$\mathbf{S}^{k+m} x = \mathbf{S}^m u,$$

which in turn becomes

$$\mathbf{S}^k t = \mathbf{S}^m u.$$

We now have a formula in which x no longer occurs, so the quantifier may be omitted.   ⊣

There are several interesting by-products of the quantifier elimination procedure. For one, we get an alternative proof of the completeness of $\mathrm{Cn}\, A_S$. Suppose we begin with a sentence σ. The quantifier elimination procedure gives a quantifier-free *sentence* τ such that (by Exercise 3) $A_S \models (\sigma \leftrightarrow \tau)$. Now we claim that either $A_S \models \tau$ or $A_S \models \neg\tau$. For τ is built up from atomic sentences by means of $\neg$ and $\rightarrow$. An atomic sentence must be of the form $\mathbf{S}^k \mathbf{0} = \mathbf{S}^l \mathbf{0}$ and is deducible from A_S if $k = l$, but is refutable (i.e., its negation is deducible) from A_S if $k \neq l$. (In fact, just {S1, S2} suffices for this.) Since every atomic sentence can be deduced or refuted, so can every quantifier-free sentence. This establishes the claim. And so either $A_S \models \sigma$ or $A_S \models \neg\sigma$.

Another by-product concerns the problem of definability in $\mathfrak{N}_S$; see Exercises 4 and 5. For any formula φ in which just v_1 and v_2 occur free we now can find a quantifier-free ψ (with the same variables free) such that

$$\mathrm{Th}\, \mathfrak{N}_S \models \forall v_1 \forall v_2 (\varphi \leftrightarrow \psi);$$

i.e.,

$$\models_{\mathfrak{N}_S} \forall v_1 \forall v_2 (\varphi \leftrightarrow \psi).$$

Thus the relation φ defined is also definable by a quantifier-free formula.

Exercises

1. Let A_S^* be the set of sentences consisting of S1, S2, and all sentences of the form

$$\varphi(\mathbf{0}) \rightarrow \forall\, v_1(\varphi(v_1) \rightarrow \varphi(\mathbf{S}v_1)) \rightarrow \forall\, v_1\, \varphi(v_1),$$

where φ is a wff (in the language of $\mathfrak{N}_S$) in which no variable except v_1 occurs free. Show that $A_S \subseteq \text{Cn}\, A_S^*$. Conclude that $\text{Cn}\, A_S^* = \text{Th}\, \mathfrak{N}_S$. (Here $\varphi(t)$ is by definition $\varphi_t^{v_1}$. The sentence displayed above is called the *induction axiom* for φ.)

2. Complete the proof of Theorem 31F. *Suggestion*: Use induction.

3. The proof of quantifier elimination for $\text{Th}\, \mathfrak{N}_S$ showed how, given a formula φ, to find a quantifier-free ψ. Show that

$$A_S \models (\varphi \leftrightarrow \psi)$$

without using the completeness of $\text{Cn}\, A_S$. (This yields an alternative proof of the completeness of $\text{Cn}\, A_S$, not involving Z-chains or the Łoś–Vaught test.)

4. Show that a subset of $\mathbb{N}$ is definable in $\mathfrak{N}_S$ iff either it is finite or its complement (in $\mathbb{N}$) is finite.

5. Show that the ordering relation $\{\langle m, n \rangle \mid m < n \text{ in } \mathbb{N}\}$ is not definable in $\mathfrak{N}_S$. *Suggestion*: It suffices to show there is no quantifier-free definition of ordering. Call a relation $R \subseteq \mathbb{N} \times \mathbb{N}$ *linear* if it can be covered by a finite number of lines. Call R *colinear* if it is the complement of a linear relation. Show that any relation definable in $\mathfrak{N}_S$ is either linear or colinear. And that the ordering relation is neither linear nor colinear.

6. Show that $\text{Th}\, \mathfrak{N}_S$ is not finitely axiomatizable. *Suggestion*: Show that no finite subset of A_S suffices, and then apply Section 2.6.

SECTION 3.2
Other Reducts of Number Theory[1]

First let us add the ordering symbol $<$ to the language. The intended structure is

$$\mathfrak{N}_L = (\mathbb{N}; 0, S, <).$$

---

[1] This section may be omitted without disastrous effects.

We want to show that the theory of this structure is (like Th $\mathfrak{N}_S$) decidable and also admits elimination of quantifiers. But unlike Th $\mathfrak{N}_S$, it is finitely axiomatizable and is not categorical in any infinite cardinality.

As axioms of Th $\mathfrak{N}_L$ we will take the finite set A_L consisting of the six sentences listed below. Here $x \leq y$ is, of course, an abbreviation for $(x < y \lor x = y)$, and $x \not\leq y$ abbreviates the negation of this formula.

$$\forall y \qquad\qquad (y \neq \mathbf{0} \rightarrow \exists x\, y = \mathbf{S}x) \qquad\qquad\qquad\text{(S3)}$$

$$\forall x \forall y \qquad (x < \mathbf{S}y \leftrightarrow x \leq y) \qquad\qquad\qquad\text{(L1)}$$

$$\forall x \qquad\qquad x \not< \mathbf{0} \qquad\qquad\qquad\qquad\qquad\quad\text{(L2)}$$

$$\forall x \forall y \qquad (x < y \lor x = y \lor y < x) \qquad\qquad\text{(L3)}$$

$$\forall x \forall y \qquad (x < y \rightarrow y \not< x) \qquad\qquad\qquad\text{(L4)}$$

$$\forall x \forall y \forall z \quad (x < y \rightarrow y < z \rightarrow x < z) \qquad\quad\text{(L5)}$$

On the one hand, it is easy to see that all six axioms are true in $\mathfrak{N}_L$. Thus Cn $A_L \subseteq$ Th $\mathfrak{N}_L$. On the other hand, the opposite inclusion is not obvious, and requires proof. We begin by listing some consequence of these axioms.

(1) $A_L \vdash \forall x\, x < \mathbf{S}x$.

PROOF.   In L1 take y to be x. $\dashv$

(2) $A_L \vdash \forall x\, x \not< x$.

PROOF.   In L4 take y to be x. $\dashv$

(3) $A_L \vdash \forall x \forall y(x \not< y \leftrightarrow y \leq x)$ (trichotomy).

PROOF.   For "$\rightarrow$" use L3. For "$\leftarrow$" use L4 and (2). $\dashv$

(4) $A_L \vdash \forall x \forall y(x < y \leftrightarrow \mathbf{S}x < \mathbf{S}y)$.

PROOF.   From A_L we can deduce the biconditionals:

$$
\begin{aligned}
x < y &\leftrightarrow y \not\leq x &&\text{by (3);}\\
&\leftrightarrow y \not< \mathbf{S}x &&\text{by L1;}\\
&\leftrightarrow \mathbf{S}x \leq y &&\text{by (3);}\\
&\leftrightarrow \mathbf{S}x < \mathbf{S}y &&\text{by L1.}
\end{aligned}
$$

$\dashv$

(5) $A_L \vdash$ S1 and $A_L \vdash$ S2.

PROOF.   S1 follows from L2 and (1). S2 comes from (4) by use of L3 and (2). $\dashv$

(6) $A_L \vdash$ S4.n for $n = 1, 2, \ldots$.

PROOF.   This follows from (1) and (2), by using L5. $\dashv$

Thus any model $\mathfrak{A}$ of A_L is (when we ignore $<^{\mathfrak{A}}$) also a model of A_S. So it must consist of a standard part plus zero or more Z-chains. In addition, it is ordered by $<^{\mathfrak{A}}$.

THEOREM 32A   The theory Cn A_L admits elimination of quantifiers.

PROOF.   Again we consider a formula

$$\exists x(\beta_0 \wedge \cdots \wedge \beta_p),$$

where each β_i is atomic or the negation of an atomic formula. The terms are, as in Section 3.1, of the form $\mathbf{S}^k u$, where u is $\mathbf{0}$ or a variable. There are two possibilities for atomic formulas,

$$\mathbf{S}^k u = \mathbf{S}^l t \quad \text{and} \quad \mathbf{S}^k u < \mathbf{S}^l t.$$

1. We can eliminate the negation symbol. Replace $t_1 \not< t_2$ by $t_2 < t_1 \vee t_1 = t_2$ and replace $t_1 \neq t_2$ by $t_1 < t_2 \vee t_2 < t_1$. (This is justified by L3 and L4.) By regrouping the atomic formulas and noting that

$$\exists x(\varphi \wedge \psi) \models\exists x \varphi \wedge \exists x \psi,$$

we may again reach formulas of the form

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_q),$$

where now each α_i is atomic.
2. We may suppose that the variable x occurs in each α_i. This is because if x does not occur in α, then

$$\exists x(\alpha \wedge \beta) \models\alpha \wedge \exists x \beta.$$

Furthermore, we may suppose that x occurs on only one side of the equality or inequality α_i. For $\mathbf{S}^k x = \mathbf{S}^l x$ can be dealt with as in Section 3.1. $\mathbf{S}^k x < \mathbf{S}^l x$ can be replaced by $\mathbf{0} = \mathbf{0}$ if $k < l$, and $\mathbf{0} \neq \mathbf{0}$ otherwise. (This is justified by L1 and L4.)

Case 1: Suppose that some α_i is an equality. Then we can proceed as in case 2 of the quantifier-elimination proof of Theorem 31G.
Case 2: Otherwise each α_i is an inequality. Then the formula can be rewritten

$$\exists x \left(\bigwedge_i t_i < \mathbf{S}^{m_i} x \wedge \bigwedge_j \mathbf{S}^{n_j} x < u_j \right).$$

(Here $\bigwedge_i$ indicates the conjunction of formulas indexed by i, so $\gamma_0 \wedge \gamma_1 \wedge \cdots \wedge \gamma_k$ can be abbreviated $\bigwedge_i \gamma_i$.) In the first conjunction, $\bigwedge_i t_i < \mathbf{S}^{m_i} x$, we have the lower bounds on x; in the second conjunction, $\bigwedge_j \mathbf{S}^{n_j} x < u_j$, we have the upper bounds. If the second conjunction is empty (i.e., if there are no upper bounds

on x), then we can replace the formula by $\mathbf{0} = \mathbf{0}$. (Why?) If the first conjunction is empty (i.e., if there are no lower bounds on x), then we can replace the formula by

$$\bigwedge_j \mathbf{S}^{n_j}\mathbf{0} < u_j,$$

which asserts that zero satisfies the upper bounds. Otherwise, we rewrite the formula successively as

$$\exists x \bigwedge_{i,j} (t_i < \mathbf{S}^{m_i} x \wedge \mathbf{S}^{n_j} x < u_j). \tag{1}$$

$$\exists x \bigwedge_{i,j} (\mathbf{S}^{n_j} t_i < \mathbf{S}^{m_i+n_j} x < \mathbf{S}^{m_i} u_j). \tag{2}$$

$$\left(\bigwedge_{i,j} \mathbf{S}^{n_j+1} t_i < \mathbf{S}^{m_i} u_j\right) \wedge \bigwedge_j \mathbf{S}^{n_j}\mathbf{0} < u_j. \tag{3}$$

This last formula says "any lower bound plus one satisfies any upper bound, and furthermore zero satisfies any upper bound." This implies that there is a gap between the greatest lower bound and the least upper bound, whence there is a solution for x. The second part guarantees that the solution for x is not forced to be negative.

In each case, we have arrived at a quantifier-free version of the given formula.   ⊣

COROLLARY 32B   (a) Cn A_L is complete.
    (b) Cn $A_L = $ Th $\mathfrak{N}_L$.
    ⋆(c) Th $\mathfrak{N}_L$ is decidable.

PROOF.   (a) The argument that followed the proof of Theorem 31G is applicable here also. (b) This follows from (a), since Cn $A_L \subseteq$ Th $\mathfrak{N}_L$ and Th $\mathfrak{N}_L$ is satisfiable. For (c), we can use the fact that any complete axiomatizable theory is decidable. But the quantifier elimination proof yields a more efficient decision procedure.   ⊣

COROLLARY 32C   A subset of $\mathbb{N}$ is definable in $\mathfrak{N}_L$ iff it is either finite or has finite complement.

PROOF.   Compare Exercise 4 of the preceding section.   ⊣

On the other hand, $\mathfrak{N}_L$ has more definable binary relations than has $\mathfrak{N}_S$. For the ordering relation $\{\langle m, n\rangle \mid m < n\}$ is not definable in $\mathfrak{N}_S$, by Exercise 5 of the preceding section.

COROLLARY 32D   The addition relation,

$$\{\langle m, n, p\rangle \mid m + n = p\},$$

is not definable in $\mathfrak{N}_L$.

PROOF.   If we could define addition, we could then define the set of even natural numbers. But this set is neither finite nor has finite complement.   ⊣

Now suppose we augment the language by the addition symbol $+$. The intended structure is

$$\mathfrak{N}_A = (\mathbb{N}; 0, S, <, +).$$

The theory of this structure is also decidable, as we will prove shortly. But to keep matters from getting even more complicated, we will avoid listing any convenient set of axioms for the theory.

The nonstandard models of Th $\mathfrak{N}_A$ must also be models of Th $\mathfrak{N}_L$. So they have a standard part, followed by some Z-chains. But ordering among the Z-chains can no longer be arbitrary. Let $\mathfrak{A}$ be a nonstandard model of Th $\mathfrak{N}_A$. The ordering $<^{\mathfrak{A}}$ induces a well-defined ordering on the set of Z-chains. (See Exercise 3.) We claim that there is no largest Z-chain, there is no smallest Z-chain, and there is between any two Z-chains another one. The reasons, in outline, can be stated simply: If a belongs to some Z-chain (i.e., is an infinite element of $\mathfrak{A}$), then $a +^{\mathfrak{A}} a$ is in a larger Z-chain. There must be some b such that $b +^{\mathfrak{A}} b$ is either a or its successor; b must be in a smaller Z-chain. If a_1 and a_2 belong to different Z-chains, then there must be some b such that $b +^{\mathfrak{A}} b$ is either $a_1 +^{\mathfrak{A}} a_2$ or its successor. And b will lie in a Z-chain between that of a_1 and that of a_2. (These statements should seem quite plausible. The reader who enjoys working with infinite numbers might supply some details.)

*THEOREM 32E (PRESBURGER, 1929)   The theory of the structure $\mathfrak{N}_A = (\mathbb{N}; 0, S, <, +)$ is decidable.

The proof is again based on a quantifier elimination procedure. The theory of $\mathfrak{N}_A$ itself does *not* admit elimination of quantifiers. For example, the formula defining the set of even numbers

$$\exists y \, v_1 = y + y$$

is not equivalent to any quantifier-free formula. We can overcome this by adding a new symbol $\equiv_2$ for congruence modulo 2. Similarly, we add symbols $\equiv_3, \equiv_4, \ldots$. The intended structure for this expanded language is

$$\mathfrak{N}^{\equiv} = (\mathbb{N}; 0, S, <, +, \equiv_2, \equiv_3, \ldots),$$

where $\equiv_k$ is the binary relation of congruence modulo k. It turns out that the theory of this structure does admit elimination of quantifiers.

This by itself does not imply that the theory of either structure is decidable. After all, we can start with *any* structure, and expand

it to a structure having additional relations until a structure is obtained that admits elimination of quantifiers. To obtain decidability, we must show that we can, given a sentence σ, (1) effectively find a quantifier-free equivalent σ', and then (2) effectively decide if σ' is true.

We will now give the quantifier elimination procedure for Th $\mathfrak{N}^{\equiv}$. For a term t and a natural number n, let nt be the term $t + t + \cdots + t$, with n summands. $0t$ is $\mathbf{0}$. Then any term can be expanded to one of the form

$$\mathbf{S}^{n_0}\mathbf{0} + n_1 x_1 + \cdots + n_k x_k$$

for $k \geq 0$, $n_i \geq 0$ (where the x_i's are variables). For example,

$$\mathbf{S}(x + \mathbf{S0}) + \mathbf{S}y$$

becomes

$$\mathbf{S}^3\mathbf{0} + x + y.$$

As usual we begin with a formula $\exists\, y(\beta_1 \wedge \cdots \wedge \beta_n)$, where β_i is an atomic formula or the negation of one.

1. Eliminate negation. Replace $\neg(t_1 = t_2)$ by $(t_1 < t_2 \vee t_2 < t_1)$. Replace $\neg(t_1 < t_2)$ by $(t_1 = t_2 \vee t_2 < t_1)$. And replace $\neg(t_1 \equiv_m t_2)$ by

$$t_1 \equiv_m t_2 + \mathbf{S}^1\mathbf{0} \vee \cdots \vee t_1 \equiv_m t_2 + \mathbf{S}^{m-1}\mathbf{0}.$$

Then regroup into a disjunction of formulas of the form

$$\exists\, y(\alpha_1 \wedge \cdots \wedge \alpha_m),$$

where each α_i is atomic. We may further suppose, as before, that y occurs in each α_i, and in fact that α_i has one of the four forms

$$
\begin{aligned}
ny + t &= u, \\
ny + t &\equiv_m u, \\
ny + t &< u, \\
u &< ny + t,
\end{aligned}
$$

where u and t are terms not containing y. In what follows we will take the liberty of writing these formulas with a subtraction symbol:

$$
\begin{aligned}
ny &= u - t, \\
ny &\equiv_m u - t, \\
ny &< u - t, \\
u - t &< ny.
\end{aligned}
$$

These are merely abbreviations for the formulas without subtraction obtained by transposing terms.

For example, we might have at this point the formula

$$\exists\, y(w < 4y \wedge 2y < u \wedge 3y < v \wedge y \equiv_3 t),$$

where t, u, v, and w are terms not containing y.

2. Uniformize the coefficients of y. Let p be the least common multiple of the coefficients of y. Each atomic formula can be converted to one in which the coefficient of y is p, by "multiplying through" by the appropriate factor. This is obviously legitimate for equalities and inequalities. In the case of congruences one must remember to raise the modulus also:

$$a \equiv_m b \quad \text{iff} \quad ka \equiv_{km} kb.$$

In the example above p is 12, and we obtain

$$\exists\, y(3w < 12y \wedge 12y < 6u \wedge 12y < 4v \wedge 12y \equiv_{36} 12t).$$

3. Eliminate the coefficient of y. Replace py by x and add the new conjunct $x \equiv_p \mathbf{0}$. (In place of $\exists\, y \cdots 12y \cdots$ we can equally well have, "There exists a multiple x of 12 such that $\cdots x \cdots$.") Our example is now converted to

$$\exists\, x(3w < x \wedge x < 6u \wedge x < 4v \wedge x \equiv_{36} 12t \wedge x \equiv_{12} \mathbf{0}).$$

4. Special case. If one of the atomic formulas is an equality, $x + t = u$, then we can replace

$$\exists\, x\, \theta$$

by

$$\theta^x_{u-t} \wedge t \le u.$$

Here replacement of x by "$u - t$" is the natural thing; we transpose terms to compensate for the absence of subtraction. For example,

$$(x \equiv_m v)^x_{u-t} \quad \text{is} \quad u \equiv_m v + t.$$

5. We may assume henceforth that $=$ does not occur. So we have a formula of the form

$$\exists\, x[r_0 - s_0 < x \wedge \cdots \wedge r_{l-1} - s_{l-1} < x$$
$$\wedge\, x < t_0 - u_0 \wedge \cdots \wedge x < t_{k-1} - u_{k-1}$$
$$\wedge\, x \equiv_{m_0} v_0 - w_0 \wedge \cdots \wedge x \equiv_{m_{n-1}} v_{n-1} - w_{n-1}],$$

where r_i, s_i, t_i, u_i, v_i, and w_i are terms not containing x. This can be abbreviated

$$\exists\, x\left[\bigwedge_{j<l} r_j - s_j < x \;\wedge\; \bigwedge_{i<k} x < t_i - u_i \;\wedge\; \bigwedge_{i<n} x \equiv_{m_i} v_i - w_i\right].$$

If there are no congruences (i.e., $n = 0$), then the formula asserts that there is a nonnegative space between the lower and

upper bounds. We can replace the formula by the quantifier-free formula:

$$\bigwedge_{i<k}\bigwedge_{j<l}(r_j - s_j) + \mathbf{S0} < t_i - u_i \wedge \bigwedge_{i<k} \mathbf{0} < t_i - u_i.$$

Let M be the least common multiple of the moduli $m_0, \ldots, m_{n-1}$. Then $a + M \equiv_{m_i} a$. So as a increases, the pattern of residues of a modulo $m_0, \ldots, m_{n-1}$ has period M. Thus, in searching for a solution to the congruences, we need only search M consecutive integers.

We now have a formula that asserts the existence of a natural number which is not less than certain lower bounds $L_1, \ldots, L_l$ and which satisfies certain upper bounds and certain congruences. If there is such a solution, then one of the following is a solution:

$$L_1, L_1 + 1, \ldots, L_1 + M - 1,$$
$$L_2, L_2 + 1, \ldots, L_2 + M - 1,$$
$$\ldots$$
$$L_l, L_l + 1, \ldots, L_l + M - 1,$$
$$0, 1, \ldots, M - 1.$$

(The last line is needed to cover the case in which every L_j is negative. To avoid having to treat this line as a special case, we will add a new lower bound of 0. That is, let $r_l = \mathbf{0}$ and $s_l = \mathbf{S0}$ so that

$$r_l - s_l < x$$

is a formula $\mathbf{0} < x + \mathbf{S0}$ asserting that x is nonnegative. We now have $l + 1$ lower bounds.)

Our formula (asserting the existence of a solution for x) can now be replaced by a quantifier-free disjunction that asserts that one of the numbers in the above matrix is a nonnegative solution:

$$\bigvee_{j\leq l}\bigvee_{1\leq q\leq M}\left[\bigwedge_{i\leq l} r_i - s_i < (r_j - s_j) + \mathbf{S}^q\mathbf{0}\right.$$
$$\wedge \bigwedge_{i<k}(r_j - s_j) + \mathbf{S}^q\mathbf{0} < t_i - u_i$$
$$\left. \wedge \bigwedge_{i<n}(r_j - s_j) + \mathbf{S}^q\mathbf{0} \equiv_{m_i} v_i - w_i\right].$$

In our continuing example we have, after adding the new lower bound on x,

$$\exists x(3w < x \wedge \mathbf{0} < x + \mathbf{S0} \wedge x < 6u \wedge x < 4v$$
$$\wedge x \equiv_{36} 12t \wedge x \equiv_{12} \mathbf{0}).$$

The quantifier-free equivalent is a disjunction of 72 conjunctions. Each conjunction has six constituents.

This proves half of the theorem. If we are given a sentence σ, the above procedure tells us how to find effectively a quantifier-free sentence τ (in the language of $\mathfrak{N}^=$) that is true (in the intended structure) iff σ is. Now we must decide if τ is true.

But this is easy. It is enough to look at atomic sentences. Any variable-free term can be put in the form $\mathbf{S}^n\mathbf{0}$. Then, for example,

$$\mathbf{S}^n\mathbf{0} \equiv_m \mathbf{S}^p\mathbf{0}$$

is true iff $n \equiv_m p$.   ⊣

Thus we have a decision procedure for Th $\mathfrak{N}_A$. In 1974 Michael Fischer and Michael Rabin showed, however, that there is no decision procedure that is fast enough to be feasible for very long formulas.

A set D of natural numbers is said to be *periodic* if for some positive p, any number n is in D iff $n + p$ is in D. D is *eventually periodic* iff there exist positive numbers M and p such that for all n greater than M, $n \in D$ iff $n + p \in D$.

THEOREM 32F   A set of natural numbers is definable in $(\mathbb{N}; 0, S, <, +)$ iff it is eventually periodic.

PROOF.   Exercise 1 asserts that every eventually periodic set is definable. Conversely, suppose D is definable. Then D is definable in $\mathfrak{N}^=$ by a quantifier-free formula (whose only variable is v_1). Since the class of eventually periodic sets is closed under union, intersection, and complementation, it suffices to show that every atomic formula in the language of $\mathfrak{N}^=$ whose only variable is v_1 defines an eventually periodic set. There are only four possibilities:

$$\begin{aligned} nv_1 + t &= u, \\ nv_1 + t &< u, \\ u &< nv_1 + t, \\ nv_1 + t &\equiv_m u, \end{aligned}$$

where u and t are numerals. The first two formulas define finite sets (which eventually have period 1), the third defines a set with finite complement, and the last formula defines a periodic set with period m.   ⊣

COROLLARY 32G   The multiplication relation

$$\{\langle m, n, p \rangle \mid p = m \cdot n \text{ in } \mathbb{N}\}$$

is not definable in $(\mathbb{N}; 0, S, <, +)$.

PROOF.   If we had a definition of multiplication, we could then use that to define the set of squares. But the set of squares is not eventually periodic.   ⊣

Exercises

1. Show that any eventually periodic set of natural numbers is definable in the structure $\mathfrak{N}_A$.

2. Show that in the structure $(\mathbb{N}; +)$ the following relations are definable:
   (a) Ordering, $\{\langle m, n \rangle \mid m < n\}$.
   (b) Zero, $\{0\}$.
   (c) Successor, $\{\langle m, n \rangle \mid n = S(m)\}$.

3. Let $\mathfrak{A}$ be a model of $\mathrm{Th}\,\mathfrak{N}_L$ (or equivalently a model of A_L). For a and b in $|\mathfrak{A}|$ define the equivalence relation:

$$a \sim b \iff \textbf{S}^{\mathfrak{A}} \text{ can be applied a finite number of times to one of } a, b \text{ to reach the other.}$$

   Let $[a]$ be the equivalence class to which a belongs. Order equivalence classes by

$$[a] \prec [b] \quad \text{iff} \quad a <^{\mathfrak{A}} b \quad \text{and} \quad a \not\sim b.$$

   Show that this is a well-defined ordering on the set of equivalence classes.

4. Show that the theory of the real numbers with its usual ordering, $\mathrm{Th}(\mathbb{R}; <)$, admits elimination of quantifiers. (Assume that the language includes equality.)

SECTION 3.3
A Subtheory of Number Theory

We now return to the full language of number theory, as described in Section 3.0. The parameters of the language are $\forall, \mathbf{0}, \mathbf{S}, <, +, \cdot$, and $\mathbf{E}$. The intended structure for this language is

$$\mathfrak{N} = (\mathbb{N}; 0, S, <, +, \cdot, E).$$

Actually in $(\mathbb{N}; \cdot, E)$ we can define $\{0\}$, S, $<$, and $+$. (See Exercise 1.) As we will show in Section 3.8, in $(\mathbb{N}; +, \cdot)$ we can define E as well as 0, S, and $<$. So there are ways in which we could economize. The luxury of having all these parameters (particularly $\mathbf{E}$) will simplify some of the proofs.

As we shall see, $\mathrm{Th}\,\mathfrak{N}$ is a very strong theory and is neither decidable nor axiomatizable. In order to prove this fact (and a number of related results), it will be strategically wise to select for study a finitely axiomatizable subtheory of $\mathrm{Th}\,\mathfrak{N}$. As hinted at in Section 3.0, this subtheory should be strong enough to represent (in a sense to be made precise) facts about decidable sets. The subtheory we have selected is $\mathrm{Cn}\,A_E$,

where A_E is the set consisting of the eleven sentences listed below. (As in the preceding section, $x \leq y$ abbreviates $x < y \lor x = y$.)

Set A_E of Axioms

$$\forall x \qquad \mathbf{S}x \neq \mathbf{0} \tag{S1}$$

$$\forall x \, \forall y \quad (\mathbf{S}x = \mathbf{S}y \rightarrow x = y) \tag{S2}$$

$$\forall x \, \forall y \quad (x < \mathbf{S}y \leftrightarrow x \leq y) \tag{L1}$$

$$\forall x \qquad x \not< \mathbf{0} \tag{L2}$$

$$\forall x \, \forall y \quad (x < y \lor x = y \lor y < x) \tag{L3}$$

$$\forall x \qquad x + \mathbf{0} = x \tag{A1}$$

$$\forall x \, \forall y \quad x + \mathbf{S}y = \mathbf{S}(x + y) \tag{A2}$$

$$\forall x \qquad x \cdot \mathbf{0} = \mathbf{0} \tag{M1}$$

$$\forall x \, \forall y \quad x \cdot \mathbf{S}y = x \cdot y + x \tag{M2}$$

$$\forall x \qquad x \, \mathbf{E} \mathbf{0} = \mathbf{S} \mathbf{0} \tag{E1}$$

$$\forall x \, \forall y \quad x \, \mathbf{E} \, \mathbf{S}y = x \, \mathbf{E} \, y \cdot x \tag{E2}$$

Since $\mathfrak{N}$ is a model of A_E, we have $\mathrm{Cn} \, A_E \subseteq \mathrm{Th} \, \mathfrak{N}$. But (as we will prove in Section 3.5) equality does not hold here. In fact, it can be shown that $A_E \nvdash \mathrm{S3}$, where S3 is the sentence $\forall y(y \neq \mathbf{0} \rightarrow \exists x \, y = \mathbf{S}x)$.

The first five axioms give us *some*, but not all, of the axioms regarding $\mathbf{S}$ and $<$ that were useful in the preceding sections. The other six axioms are the "recursion" equations for addition, multiplication, and exponentiation.

We first show that certain simple sentences in $\mathrm{Th} \, \mathfrak{N}$ are deducible from A_E.

LEMMA 33A   (a) $A_E \vdash \forall x \, x \not< \mathbf{0}$.
(b) For any natural number k,

$$A_E \vdash \forall x(x < \mathbf{S}^{k+1}\mathbf{0} \leftrightarrow x = \mathbf{S}^0\mathbf{0} \lor \cdots \lor x = \mathbf{S}^k\mathbf{0}).$$

Notice that (a) can be thought of as the $k = -1$ case of (b), where the empty disjunction is $\bot$. The lemma tells us that A_E "knows" that the numbers less than 7, for example, are exactly 0, 1, 2, 3, 4, 5, 6. So in any model of A_E, the standard points — the ones denoted by numerals $S^k\mathbf{0}$ — are ordered in the natural way, and (by L3) the infinite points, if any, are all larger than any standard point.

PROOF.   Part (a) is L2. For (b) we use induction (in English) on k. We have as a consequence of L1,

$$x < \mathbf{S}\mathbf{0} \leftrightarrow x < \mathbf{0} \lor x = \mathbf{0},$$

which together with L2 gives

$$x < \mathbf{S0} \leftrightarrow x = \mathbf{0},$$

which is the $k = 0$ case of (b). For the inductive step we again apply L1:

$$x < \mathbf{S}^{k+1}\mathbf{0} \leftrightarrow x < \mathbf{S}^k\mathbf{0} \vee x = \mathbf{S}^k\mathbf{0}.$$

By the inductive hypothesis, $x < \mathbf{S}^k\mathbf{0}$ can be replaced by

$$x = \mathbf{S}^0\mathbf{0} \vee \cdots \vee x = \mathbf{S}^{k-1}\mathbf{0},$$

whereby we obtain (b). ⊣

LEMMA 33B   For any variable-free term t, there is a unique natural number n such that

$$A_E \vdash t = \mathbf{S}^n\mathbf{0}.$$

PROOF.   The uniqueness is immediate. (Why? Because A_E, weak as it is, at least knows, by S1, that $7 \neq 0$, and by S2 80 times, that $87 \neq 80$.) For the existence half, we use induction on t. If t is $\mathbf{0}$, we take $n = 0$. If t is $\mathbf{S}u$, then by the inductive hypothesis $A_E \vdash u = \mathbf{S}^m\mathbf{0}$ for some m. Hence $A_E \vdash t = \mathbf{S}^{m+1}\mathbf{0}$.

  Now suppose t is $u_1 + u_2$. By the inductive hypothesis $A_E \vdash t = \mathbf{S}^m\mathbf{0} + \mathbf{S}^n\mathbf{0}$ for some m and n. We now apply A2 n times and A1 once to obtain $A_E \vdash t = \mathbf{S}^{m+n}\mathbf{0}$. The arguments for multiplication and exponentiation are similar. ⊣

  As a special case of this lemma we have "$2 + 2 = 4$" (i.e., $\mathbf{S}^2\mathbf{0} + \mathbf{S}^2\mathbf{0} = \mathbf{S}^4\mathbf{0}$) as a consequence of A_E. A_E is at least smart enough to evaluate variable-free terms. And the proof shows more than this. The proof provides exact instructions for how, given such a term t, to find effectively the unique number n such that $A_E \vdash t = \mathbf{S}^n\mathbf{0}$.

THEOREM 33C   For any quantifier-free sentence τ true in $\mathfrak{N}$, $A_E \vdash \tau$.

PROOF.   Exercise 2. Start with the atomic sentences; these will be of the form $t_1 = t_2$ or $t_1 < t_2$ for variable-free terms t_1 and t_2. Show that A_E proves τ if τ is true in $\mathfrak{N}$, and refutes τ (i.e., proves $\neg \tau$) if τ is false in $\mathfrak{N}$. ⊣

  Later on, we will improve on Theorem 33C by allowing τ to contain "bounded quantifiers"; see Theorem 33I.

  A simplified notation (used earlier in Section 2.7) for substitution will be helpful in the coming pages:

$$\varphi(t) = \varphi_t^{v_1},$$
$$\varphi(t_1, t_2) = \left(\varphi_{t_1}^{v_1}\right)_{t_2}^{v_2},$$

and so forth. Thus $\varphi = \varphi(v_1) = \varphi(v_1, v_2)$. Usually the term substituted

will be a numeral, for example

$$\varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) = \left(\varphi_{\mathbf{S}^a\mathbf{0}}^{v_1}\right)_{\mathbf{S}^b\mathbf{0}}^{v_2}.$$

But at times we will also substitute other terms, e.g., $\varphi(x) = \varphi_x^{v_1}$, where x is a variable. If, however, x is not substitutable for v_1 in φ, then we must take $\varphi(x) = \psi_x^{v_1}$, where ψ is a suitable alphabetic variant of φ.

In the next proof (and elsewhere in this chapter) we make use of the following consequence of the substitution lemma of Section 2.5: For a formula φ in which at most $v_1, \ldots, v_n$ occur free and for natural numbers $a_1, \ldots, a_n$,

$$\models_{\mathfrak{N}} \varphi[\![a_1, \ldots, a_n]\!] \Leftrightarrow \models_{\mathfrak{N}} \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}).$$

An existential ($\exists_1$) formula is one of the form $\exists x_1 \cdots \exists x_k \theta$, where θ is quantifier-free. The following result improves Theorem 33C:

> **COROLLARY 33D**   If τ is an existential sentence true in $\mathfrak{N}$, then $A_E \vdash \tau$.

> PROOF.   If $\exists v_1 \exists v_2 \theta$ is true in $\mathfrak{N}$, then for some natural numbers m and n, $\theta(\mathbf{S}^m\mathbf{0}, \mathbf{S}^n\mathbf{0})$ is true in $\mathfrak{N}$. As this is a quantifier-free true sentence, it is deducible from A_E. But it in turn logically implies $\exists v_1 \exists v_2 \theta$. $\dashv$

On the other hand, it is known that there are true universal ($\forall_1$) sentences (i.e., of the form $\forall x_1 \cdots \forall x_k \theta$ for quantifier-free θ) that are *not* in Cn A_E.

Representable Relations

Let R be an m-ary relation on $\mathbb{N}$; i.e., $R \subseteq \mathbb{N}^m$. We know that a formula ρ (in which only $v_1, \ldots, v_m$ occur free) defines R in $\mathfrak{N}$ iff for every $a_1, \ldots, a_m$ in $\mathbb{N}$,

$$\langle a_1, \ldots, a_m \rangle \in R \Leftrightarrow \models_{\mathfrak{N}} \rho[\![a_1, \ldots, a_m]\!]$$
$$\Leftrightarrow \models_{\mathfrak{N}} \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}).$$

(The last condition here is equivalent to the preceding one by the substitution lemma.) We can recast this into two implications:

$$\langle a_1, \ldots, a_m \rangle \in R \Rightarrow \models_{\mathfrak{N}} \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}),$$
$$\langle a_1, \ldots, a_m \rangle \notin R \Rightarrow \models_{\mathfrak{N}} \neg \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}).$$

We will say that ρ also represents R in the theory Cn A_E if in these two implications the notion of truth in $\mathfrak{N}$ can be replaced by the stronger notion of deducibility from A_E.

More generally, let T be any theory in a language with $\mathbf{0}$ and $\mathbf{S}$. Then ρ *represents* R in T iff for every $a_1, \ldots, a_m$ in $\mathbb{N}$:

$$\langle a_1, \ldots, a_m \rangle \in R \Rightarrow \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) \in T,$$
$$\langle a_1, \ldots, a_m \rangle \notin R \Rightarrow (\neg \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})) \in T.$$

For example, ρ represents R in the theory Th $\mathfrak{N}$ iff ρ defines R in $\mathfrak{N}$. But ρ represents R in Cn A_E iff for all $a_1, \ldots, a_m$:

$$\langle a_1, \ldots, a_m \rangle \in R \Rightarrow A_E \vdash \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}),$$
$$\langle a_1, \ldots, a_m \rangle \notin R \Rightarrow A_E \vdash \neg\, \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}).$$

The equality relation on $\mathbb{N}$, for example, is represented in Cn A_E by the formula $v_1 = v_2$. For

$$m = n \Rightarrow\, \vdash \mathbf{S}^m\mathbf{0} = \mathbf{S}^n\mathbf{0},$$
$$m = n \Rightarrow \{\text{S1, S2}\} \vdash \neg\, \mathbf{S}^m\mathbf{0} = \mathbf{S}^n\mathbf{0}.$$

A relation is *representable* in T iff there exists some formula that represents it in T.

The concept of representability should be compared with that of definability. In both cases we are somehow describing relations on the natural numbers by formulas. In the case of definability, we ask about the *truth* of sentences in the interpretation. In the case of representability in Cn A_E, we ask instead about the *deducibility* of sentences from the axioms.

Say that a formula φ, in which no variables other than $v_1, \ldots, v_m$ occur free, is *numeralwise determined* by A_E iff for every m-tuple $a_1, \ldots, a_m$ of natural numbers, either

$$A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})$$

or

$$A_E \vdash \neg\, \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})$$

THEOREM 33E   A formula ρ represents a relation R in Cn A_E iff

(1) ρ is numeralwise determined by A_E, and
(2) ρ defines R in $\mathfrak{N}$.

PROOF.   We use the fact that $\mathfrak{N}$ is a model of A_E. If ρ represents R in Cn A_E, then it is clear that (1) holds; (2) holds since "$A_E \vdash$" implies "$\models_{\mathfrak{N}}$." Conversely, if (1) and (2) hold, then we have

$$\langle a_1, \ldots, a_m \rangle \in R \Rightarrow \models_{\mathfrak{N}} \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) \qquad \text{by (2)}$$
$$\Rightarrow A_E \nvdash \neg\, \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) \qquad \text{since } \mathfrak{N} \text{ is a model}$$
$$\text{of } A_E$$
$$\Rightarrow A_E \vdash \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) \qquad \text{by (1).}$$

Similarly for the complement of R and $\neg\, \rho$. $\dashv$

Church's Thesis

We now turn to the relationship of the concepts of representability and decidability.

*THEOREM 33F   Assume that R is a relation representable in a consistent axiomatizable theory. Then R is decidable.

PROOF.   Say that ρ represents R in the consistent axiomatizable theory T. Recall that T is effectively enumerable (Corollary 25F). The decision procedure is as follows:

Given $a_1, \ldots, a_m$, enumerate the members of T. If, in the enumeration, $\rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})$ is found, then we are done and $\langle a_1, \ldots, a_m \rangle \in R$. If, in the enumeration, $\neg \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})$ is found, then we are done and $\langle a_1, \ldots, a_m \rangle \notin R$.

By the representability, one sentence or the other always appears eventually, whereupon the procedure terminates. Since T is consistent, the answer given by the procedure is correct.   ⊣

*COROLLARY 33G   Any relation representable in a consistent finitely axiomatizable theory is decidable.

What about the converse to the above corollary? We cannot really hope to *prove* the converse on the basis of our informal notion of decidability. For our informal approach is usable only for giving lower bounds on the class of decidable relations (i.e., for showing that certain relations *are* decidable) and is unsuited to giving upper bounds (i.e., for showing *undecidability*).

It is nevertheless possible to make plausibility arguments in support of the converse. This will be easier to do at the end of Section 3.4 than here. Roughly, the idea is that in a finite number of axioms we could capture the (finitely long) instructions for the decision procedure.

The assertion that both the above corollary and its converse are correct is generally known as *Church's thesis*. This assertion is not really a mathematical statement susceptible to proof or disproof; rather it is a judgment that the correct formalization of the informal notion of decidability is by means of representability in consistent and finitely axiomatizable theories.

DEFINITION.   A relation R on the natural numbers is *recursive* iff it is representable in some consistent finitely axiomatizable theory (in a language with $\mathbf{0}$ and $\mathbf{S}$).

Church's thesis now can be put more succinctly: A relation is decidable iff it is recursive. Or perhaps more accurately: The concept of recursiveness is the correct precise counterpart to the informal concept of decidability. The situation is analogous to one encountered in calculus. An intuitively continuous function (defined on an interval) is one whose graph you can draw without lifting your pencil off the paper. But to prove theorems, some formal counterpart of this notion is needed. And so one gives the usual definition of ε-δ-continuity. One should ask if the precise notion of ε-δ-continuity is an accurate formalization of intuitive continuity. If anything, the class of ε-δ-continuous functions is too broad. It includes nowhere differentiable functions, whose graphs cannot be drawn without lifting the pencil. But accurate or not, the class

of ε-δ-continuous functions has been found to be a natural and important class in mathematical analysis.

Very much the same situation occurs with recursiveness. One should ask if the precise notion of recursiveness is an accurate formalization of the informal notion of decidability. Again, the precisely defined class (of recursive relations) appears to be, if anything, too broad. It includes relations for which any decision procedure will, for large inputs, require so much computing time and memory ("scratchpad") space as to make implementation absurd. Recursiveness corresponds to decidability in an idealized world, where length of computation and amount of memory are disregarded. But in any case, the class of recursive relations has been found to be a natural and important class in mathematical logic.

Empirical evidence that the class of recursive relations is not too narrow is provided by the following:

1. Any relation considered thus far that mathematicians have felt was decidable has been found to be recursive.

2. Several people have tried giving precise definitions of idealized computing agents. The best-known such idealized agents are the "Turing machines," introduced by Alan Turing in 1936. (A variation on that idea leads to the register machines described in Section 3.6.) The idea was to devise something that could carry out any effective procedure. In all cases, the class of relations having decision procedures executable by such a computing agent has been exactly the class of recursive relations. (Because of the importance of Turing's analysis of effective computability, Church's thesis is often called the *Church–Turing thesis*.)

The fact that so many different (yet equivalent) definitions for the class of recursive relations have been found is some indication of the naturalness and importance of the concept.

In this book we will continue to exclude the informal notion of decidability from nonstarred theorems. But in the remainder of the exposition we will accept Church's thesis. For example, we will speak of a set's being undecidable when we have a theorem stating it to be nonrecursive.

Obviously any relation representable in Cn A_E is recursive. We will prove later that the converse also holds; if a relation is representable in any consistent finitely axiomatizable theory, then it is representable in the one theory we have selected for special study. (This was, of course, a motivating factor in our selection.)

The use of the word "recursive" in this context is the result of historical accident — even of historical error. Recently several mathematicians have argued that the word "computable" would more accurately reflect the intended ideas. But in the present context, we want to reserve the word "computable" for an informal concept, to be defined next. For relations we have the informal concept of decidability; for functions the

analogous concept is computability. (As notational shorthand, a string $a_1, \ldots, a_k$ can be written as $\vec{a}$.)

> ★DEFINITION. A function $f : \mathbb{N}^k \to \mathbb{N}$ is *computable* iff there is an effective procedure that, given any k-tuple $\vec{a}$ of natural numbers, will produce $f(\vec{a})$.

For example, addition and multiplication are computable. Effective procedures, using base-10 notation, for these functions are taught in the elementary schools. (Strictly speaking, in the concept of computability one should refer to being given *numerals*, not numbers. For it is numerals — strings of symbols like the triple 317 or the triple XCI — that can be communicated. Nonetheless, we will suppress this point.) On the other hand, of the uncountably many functions from $\mathbb{N}^k$ into $\mathbb{N}$, only countably many can be computable, because there are only countably many effective procedures.

We want to give a mathematical counterpart to the informal concept of computability, just as in the case of decidable relations. The clue to the correct counterpart is provided by the next theorem. Recall that any function $f : \mathbb{N}^k \to \mathbb{N}$ is also a $(k+1)$-ary relation on $\mathbb{N}$:

$$\langle a_1, \ldots, a_k, b \rangle \in f \quad \Longleftrightarrow \quad f(a_1, \ldots, a_k) = b.$$

At one time it was popular to distinguish between the function and the relation (which was called the *graph* of the function). Current set-theoretic usage takes a function to be the same thing as its graph. But we still have the two ways of looking at the function.

> ★THEOREM 33H The following three conditions on a function $f : \mathbb{N}^k \to \mathbb{N}$ are equivalent:
>
> (a) f is computable.
> (b) When viewed as a relation, f is a decidable relation.
> (c) When viewed as a relation, f is an effectively enumerable relation.

> PROOF. (a) $\Rightarrow$ (b): Assume that f is computable; we will describe the decision procedure. Given $\langle a_1, \ldots, a_k, b \rangle$, first compute $f(a_1, \ldots, a_k)$. Then look to see if the result is equal to b. If it is say "yes," otherwise say "no."
> (b) $\Rightarrow$ (c): Any decidable relation is effectively enumerable. For we can enumerate the set of all $(k+1)$-tuples of numbers, and place on the output list those which meet the test of belonging to the relation.
> (c) $\Rightarrow$ (a): Assume that we have an effective enumeration of (the graph of) f. To compute $f(a_1, \ldots, a_k)$ we examine the $(k+1)$-tuples in the enumeration until we find the one that begins with $a_1 \ldots, a_k$. Its last component is then the desired function value.   ⊣

Thus by using Church's thesis, we can say that f is computable iff f (viewed as a relation) is recursive. The class of recursive functions is an interesting class even apart from its connection with incompleteness theorems of logic. It represents an upper bound to the class of functions that can actually be computed by programs for digital computers. The recursive functions are those which are calculable by digital computers, provided one ignores practical limitations on computing time and memory space.

We can now describe our plans for this section and the next. Our basic goal is to obtain the theorems of Section 3.5. But some groundwork is required before we can prove those theorems; we must verify that a number of relations (intuitively decidable) and a number of functions (intuitively computable) are representable in $\text{Cn}\, A_E$ and hence are recursive. In the process we will show (Theorem 34A) that recursiveness is equivalent to representability in $\text{Cn}\, A_E$. In the remainder of the present section we will establish general facts about representability, and will show, for example, that certain functions for encoding finite sequences of numbers into single numbers are representable. Then in Section 3.4 we apply these results to particular relations and functions related to the syntactical features of the formal language.

The author is sufficiently realistic to know that many readers will be more interested in the theorems of Section 3.5 than in the preliminary spadework. If the reader is willing to believe that intuitively decidable relations are all representable in $\text{Cn}\, A_E$, and intuitively computable functions are functionally representable (a concept we will define shortly) there, then most if not all of the proofs in this spadework become unnecessary. But it is hoped that the definitions and the statements of the results will still receive some attention.

Numeralwise Determined Formulas

Theorem 33E tells us that we can show a relation to be representable in $\text{Cn}\, A_E$ by finding a formula that defines it in $\mathfrak{N}$ and is numeralwise determined by A_E. The next theorem will be useful in establishing numeralwise determination.

THEOREM 33I (a) Any atomic formula is numeralwise determined by A_E.

(b) If φ and ψ are numeralwise determined by A_E, then so are $\neg\,\varphi$ and $\varphi \rightarrow \psi$.

(c) If φ is numeralwise determined by A_E, then so are the following formulas (obtained from φ by "bounded quantification"):

$$\forall x(x < y \rightarrow \varphi),$$
$$\exists x(x < y \wedge \varphi).$$

PROOF.   Part (a) follows from Theorem 33C. Part (b) is easy. It remains to prove part (c). We will consider a formula

$$\exists x(x < y \land \varphi(x, y, z))$$

in which just the variables y and z occur free. Consider two natural numbers a and b; we must show that either

$$A_E \vdash \exists x(x < \mathbf{S}^a \mathbf{0} \land \varphi(x, \mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0}))$$

or

$$A_E \vdash \neg \exists x(x < \mathbf{S}^a \mathbf{0} \land \varphi(x, \mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0})).$$

Case 1: For some c less than a,

$$A_E \vdash \varphi(\mathbf{S}^c \mathbf{0}, \mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0}). \tag{1}$$

(This case occurs iff $\exists x(x < \mathbf{S}^a \mathbf{0} \land \varphi(x, \mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0}))$ is true in $\mathfrak{N}$.) We also have

$$A_E \vdash \mathbf{S}^c \mathbf{0} < \mathbf{S}^a \mathbf{0}. \tag{2}$$

And the sentences in (1) and (2) logically imply the sentence

$$\exists x(x < \mathbf{S}^a \mathbf{0} \land \varphi(x, \mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0})).$$

Case 2: Otherwise for every c less than a,

$$A_E \vdash \neg \varphi(\mathbf{S}^c \mathbf{0}, \mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0}). \tag{3}$$

(This case occurs iff $\forall x(x < \mathbf{S}^a \mathbf{0} \to \neg \varphi(x, \mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0}))$ is true in $\mathfrak{N}$.) We know from Lemma 33A that

$$A_E \vdash \forall x(x < \mathbf{S}^a \mathbf{0} \to x = \mathbf{S}^0 \mathbf{0} \lor \cdots \lor x = \mathbf{S}^{a-1} \mathbf{0}). \tag{4}$$

The sentence in (4) together with the sentences in (3) (for $c = 0, \ldots, a - 1$) logically imply

$$\forall x(x < \mathbf{S}^a \mathbf{0} \to \neg \varphi(x, \mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0})).$$

And this is equivalent to

$$\neg \exists x(x < \mathbf{S}^a \mathbf{0} \land \varphi(x, \mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0})).$$

This shows that $\exists x(x < y \land \varphi(x, y, z))$ is numeralwise determined by A_E. By applying this result to $\neg \varphi$ we obtain the fact that the dual formula, $\forall x(x < y \to \varphi(x, y, z))$, is numeralwise determined by A_E as well.   ⊣

The argument in case 2 relied on the fact that the x quantifier was bounded by $\mathbf{S}^a \mathbf{0}$. We will see that it is possible for

$$\neg \psi(\mathbf{S}^0 \mathbf{0}), \neg \psi(\mathbf{S}^1 \mathbf{0}), \ldots$$

all to be consequences of A_E without having

$$\forall x \, \neg \, \psi(x)$$

be a consequence.

The preceding theorem is a useful tool for showing many relations to be representable in Cn A_E. For example, the set of primes is represented by

$$\mathbf{S}^1 \mathbf{0} < v_1 \wedge \forall x (x < v_1 \rightarrow \forall y (y < v_1 \rightarrow x \cdot y \neq v_1)).$$

This formula defines the primes in $\mathfrak{N}$, and by the preceding theorem is numeralwise determined by A_E. It therefore represents the set of primes in Cn A_E.

Representable Functions

Often it is more convenient to work with functions than with relations. Let $f : \mathbb{N}^m \rightarrow \mathbb{N}$ be an m-place function on the natural numbers. A formula φ in which only $v_1, \ldots, v_{m+1}$ occur free will be said to *functionally represent* f (in the theory Cn A_E) iff for every $a_1, \ldots, a_m$ in $\mathbb{N}$,

$$A_E \vdash \forall \, v_{m+1} \big[\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, v_{m+1}) \leftrightarrow v_{m+1} = \mathbf{S}^{f(a_1,\ldots,a_m)}\mathbf{0} \big].$$

(Observe that the "$\leftarrow$" half of this sentence is equivalent to $\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots,$ $\mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{f(a_1,\ldots,a_m)}\mathbf{0})$. The "$\rightarrow$" half adds an assertion of uniqueness.)

> **Theorem 33J** If φ functionally represents f in Cn A_E, then it also represents f (as a relation) in Cn A_E.
>
> **Proof, with $m = 1$.**   Since φ functionally represents f, we have for any b:
>
> $$A_E \vdash \varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) \leftrightarrow \mathbf{S}^b\mathbf{0} = \mathbf{S}^{f(a)}\mathbf{0}.$$
>
> If $\langle a, b \rangle \in f$, i.e., if $f(a) = b$, then the right half of this biconditional is valid and we get
>
> $$A_E \vdash \varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}).$$
>
> But otherwise the right half is refutable from A_E (i.e., its negation is deducible), whence
>
> $$A_E \vdash \neg\,\varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}). \qquad\qquad \dashv$$

The converse of this theorem is false. But we can change the formula:

> **Theorem 33K** Let f be a function on $\mathbb{N}$ that is (as a relation) representable in Cn A_E. Then we can find a formula φ that functionally represents f in Cn A_E.

Proof.   To simplify the notation we will take f to be a one-place
function on $\mathbb{N}$. The desired sentence,

$$\forall\, v_2[\varphi(\mathbf{S}^a\mathbf{0},\, v_2) \leftrightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}],$$

is equivalent to the conjunction of the two sentences

$$\varphi(\mathbf{S}^a\mathbf{0},\, \mathbf{S}^{f(a)}\mathbf{0}) \tag{1}$$

and

$$\forall\, v_2[\varphi(\mathbf{S}^a\mathbf{0},\, v_2) \rightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}]. \tag{2}$$

The sentence (1) is a theorem of A_E whenever φ represents f.
The sentence (2) is an assertion of uniqueness; we must construct
φ in such a way that this will also be a theorem of A_E.

Begin with a formula θ known to represent f (as a binary
relation). Let φ be

$$\theta(v_1,\, v_2) \wedge \forall\, z(z < v_2 \rightarrow \neg\,\theta(v_1,\, z)).$$

We can then rewrite (2) as

$$\forall\, v_2[\theta(\mathbf{S}^a\mathbf{0},\, v_2) \wedge \forall\, z(z < v_2 \rightarrow \neg\,\theta(\mathbf{S}^a\mathbf{0},\, z))$$
$$\rightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}]. \tag{2$'$}$$

To show this to be a theorem of A_E it clearly suffices to show that

$$A_E \cup \{\theta(\mathbf{S}^a\mathbf{0},\, v_2),\, \forall\, z(z < v_2 \rightarrow \neg\,\theta(\mathbf{S}^a\mathbf{0},\, z))\} \vdash v_2 = \mathbf{S}^{f(a)}\mathbf{0}.$$

Call this set of hypotheses (to the left of "$\vdash$") Γ. Since L3 $\in A_E$
it suffices to show that

$$\Gamma \vdash v_2 \not< \mathbf{S}^{f(a)}\mathbf{0} \tag{3}$$

and

$$\Gamma \vdash \mathbf{S}^{f(a)}\mathbf{0} \not< v_2. \tag{4}$$

It is easy to obtain (4), since from the last member of Γ we get

$$\mathbf{S}^{f(a)}\mathbf{0} < v_2 \rightarrow \neg\,\theta(\mathbf{S}^a\mathbf{0},\, \mathbf{S}^{f(a)}\mathbf{0})$$

and we know that

$$A_E \vdash \theta(\mathbf{S}^a\mathbf{0},\, \mathbf{S}^{f(a)}\mathbf{0}). \tag{5}$$

To obtain (3) we first note that we have as theorems of A_E,

$$v_2 < \mathbf{S}^{f(a)}\mathbf{0} \leftrightarrow v_2 = \mathbf{S}^0\mathbf{0} \vee \cdots \vee v_2 = \mathbf{S}^{f(a)-1}\mathbf{0} \tag{6}$$

and

$$\neg\,\theta(\mathbf{S}^a\mathbf{0},\, \mathbf{S}^b\mathbf{0}) \quad \text{for } b = 0, \ldots, f(a) - 1. \tag{7}$$

The formulas (6) and (7) imply the formula

$$v_2 < \mathbf{S}^{f(a)}\mathbf{0} \rightarrow \neg\,\theta(\mathbf{S}^a\mathbf{0},\, v_2). \tag{8}$$

Since $\theta(\mathbf{S}^a\mathbf{0},\, v_2) \in \Gamma$, we have (3).

This shows (2) to be a theorem of A_E; (5) and (8) show (1) to be a theorem of A_E as well.   ⊣

We next want to show that certain basic functions are representable (in Cn A_E) and that the class of representable functions has certain closure properties. Henceforth in this section, when we say that a function or relation is representable, we will mean that it is representable in the theory Cn A_E. But the phrase "in Cn A_E" will usually be omitted.

In simple cases, an m-place function might be represented by an equation

$$v_{m+1} = t.$$

In fact, any such equation, when the variables in t are among $v_1, \ldots, v_m$, defines in $\mathfrak{N}$ an m-place function f. (The value of f at $\langle a_1, \ldots, a_m \rangle$ is the number assigned in $\mathfrak{N}$ to t when v_i is assigned a_i, $1 \leq i \leq m$.) Furthermore, we know that any equation is numeralwise determined by A_E, so the equation represents f as a relation. In fact, it even functionally represents f, for the sentence

$$\forall v_{m+1}[v_{m+1} = t(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) \leftrightarrow v_{m+1} = \mathbf{S}^{f(a_1,\ldots,a_m)}\mathbf{0}]$$

is logically equivalent to

$$t(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) = \mathbf{S}^{f(a_1,\ldots,a_m)}\mathbf{0},$$

which is a quantifier-free sentence true in $\mathfrak{N}$. (Here $t(u_1, \ldots, u_m)$ is the term obtained by replacing v_1 by u_1, then v_2 by u_2, etc.) For example:

1. The successor function is represented (functionally) by the equation

$$v_2 = \mathbf{S}v_1.$$

2. Any constant function is representable. The m-place function that constantly assumes the value b is represented by the equation

$$v_{m+1} = \mathbf{S}^b\mathbf{0}.$$

3. The projection function (where $1 \leq i \leq m$)

$$I_i^m(a_1, \ldots, a_m) = a_i$$

is represented by the equation

$$v_{m+1} = v_i.$$

4. Addition, multiplication, and exponentiation are represented by the equations

$$\begin{aligned}
v_3 &= v_1 + v_2, \\
v_3 &= v_1 \cdot v_2, \\
v_3 &= v_1 \mathbf{E} v_2,
\end{aligned}$$

respectively.

The reader should not be misled by these simple examples; not every representable function is representable by an equation.

We next want to show that the family of representable functions is closed under composition. To simplify the notation, we will consider a one-place function f on $\mathbb{N}$, where

$$f(a) = g(h_1(a), h_2(a)).$$

Suppose that g is functionally represented by ψ and h_i by θ_i. To represent f it would be reasonable to try either

$$\forall y_1 \forall y_2 (\theta_1(v_1, y_1) \rightarrow \theta_2(v_1, y_2) \rightarrow \psi(y_1, y_2, v_2))$$

or

$$\exists y_1 \exists y_2 (\theta_1(v_1, y_1) \wedge \theta_2(v_1, y_2) \wedge \psi(y_1, y_2, v_2)).$$

(Think of $\psi(y_1, y_2, v_2)$ as saying "$g(y_1, y_2) = v_2$" and think of $\theta_i(v_1, y_1)$ as saying "$h_i(v_1) = y_i$." Then the first formula translates, "For any y_1, y_2, if $h_1(v_1) = y_1$ and $h_2(v_1) = y_2$, then $g(y_1, y_2) = v_2$." The second formula translates, "There exist y_1, y_2 such that $h_1(v_1) = y_1$ and $h_2(v_1) = y_2$ and $g(y_1, y_2) = v_2$." Either one is a reasonable way of saying, "$g(h_1(v_1), h_2(v_1)) = v_2$." There are two choices, because when something is unique, either quantifier can be used for it.)

Actually either formula would work; let φ be

$$\forall y_1 \forall y_2 (\theta_1(v_1, y_1) \rightarrow \theta_2(v_1, y_2) \rightarrow \psi(y_1, y_2, v_2)).$$

Consider any natural number a; we have at our disposal

$$\forall v_2 [\psi(\mathbf{S}^{h_1(a)}\mathbf{0}, \mathbf{S}^{h_2(a)}\mathbf{0}, v_2) \leftrightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}]. \tag{1}$$

$$\forall y_1 [\theta_1(\mathbf{S}^a\mathbf{0}, y_1) \leftrightarrow y_1 = \mathbf{S}^{h_1(a)}\mathbf{0}]. \tag{2}$$

$$\forall y_2 [\theta_2(\mathbf{S}^a\mathbf{0}, y_2) \leftrightarrow y_2 = \mathbf{S}^{h_2(a)}\mathbf{0}]. \tag{3}$$

And we want

$$\forall v_2 (\varphi(\mathbf{S}^a\mathbf{0}, v_2) \leftrightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}), \tag{4}$$

i.e.,

$$\forall v_2 (\forall y_1 \forall y_2 [\theta_1(\mathbf{S}^a\mathbf{0}, y_1) \rightarrow \theta_2(\mathbf{S}^a\mathbf{0}, y_2) \rightarrow \psi(y_1, y_2, v_2)]$$
$$\leftrightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}). \tag{4}$$

But (1), (2), and (3) imply (4), as the reader is asked to verify in Exercise 4.

More generally we have

THEOREM 33L   Let g be an n-place function, let $h_1, \ldots, h_n$ be m-place functions, and let f be defined by

$$f(a_1, \ldots, a_m) = g(h_1(a_1, \ldots, a_m), \ldots, h_n(a_1, \ldots, a_m)).$$

From formulas functionally representing g and $h_1, \ldots, h_n$ we can find a formula that functionally represents f.

In the above proof we have $m = 1$ and $n = 2$. But the general case is proved in exactly the same way.

In order to obtain a function such as

$$f(a, b) = g(h(a), b),$$

we note that

$$f(a, b) = g\big(h\big(I_1^2(a, b)\big), I_2^2(a, b)\big).$$

The above theorem then can be applied (twice) to show that f is representable (provided that g and h are).

To facilitate discussion of functions with an arbitrary number of variables, we will use vector notation. For example, the equation in the above theorem can be written

$$f(\vec{a}) = g(h_1(\vec{a}), \ldots, h_n(\vec{a})).$$

Another important closure property of the functions representable in $\mathrm{Cn}\, A_E$ is closure under the "least-zero" operator.

THEOREM 33M   Assume that the $(m + 1)$-place function g is representable and that for every $a_1, \ldots, a_m$ there is a b such that

$$g(a_1, \ldots, a_m, b) = 0.$$

Then we can find a formula that represents the m-place function f, where

$$f(a_1, \ldots, a_m) = \text{the least } b \text{ such that } g(a_1, \ldots, a_m, b) = 0.$$

(In vector notation we can rewrite this last equation:

$$f(\vec{a}) = \text{the least } b \text{ such that } g(\vec{a}, b) = 0.$$

The traditional notation for the least-zero operator is

$$f(\vec{a}) = \mu b[g(\vec{a}, b) = 0]$$

and the operator is often called "the μ-operator.")

PROOF.   To simplify the notation we take $m = 1$; thus

$$f(a) = b \quad \text{iff} \quad g(a, b) = 0 \quad \text{and for all} \quad c < b, \ g(a, c) \neq 0.$$

If ψ represents g, then we can obtain a formula representing f (as a relation) simply by formalizing the right side of this equivalence:

$$\psi(v_1, v_2, \mathbf{0}) \wedge \forall y(y < v_2 \rightarrow \neg\, \psi(v_1, y, \mathbf{0})).$$

This formula defines (the graph of) f and is numeralwise determined by A_E. ⊣

A Catalog

We now construct a repertoire of representable (in Cn A_E) functions and relations, including in particular functions for encoding and decoding sequences.

0. As a consequence of Theorem 33I, any relation that has (in $\mathfrak{N}$) a quantifier-free definition is representable. And the class of representable relations is closed under unions, intersections, and complements. And if R is representable, then so are

$$\{\langle a_1, \ldots, a_m, b\rangle \mid \text{for all } c < b, \langle a_1, \ldots, a_m, c\rangle \in R\}$$

and

$$\{\langle a_1, \ldots, a_m, b\rangle \mid \text{for some } c < b, \langle a_1, \ldots, a_m, c\rangle \in R\}.$$

For example, any finite relation has a quantifier-free definition, as does the ordering relation.

1. A relation R is representable iff its characteristic function K_R is. (K_R is the function for which $K_R(\vec{a}) = 1$ when $\vec{a} \in R$, and $K_R(\vec{a}) = 0$ otherwise.)

> PROOF.   (⟸) Say that R is a unary relation (a subset of $\mathbb{N}$) and that K_R is represented by $\psi(v_1, v_2)$. We claim that $\psi(v_1, \mathbf{S0})$ represents R. For it defines R and is numeralwise determined by A_E.
>   (⟹) Say that $\varphi(v_1)$ represents R. Then
>
> $$(\varphi(v_1) \wedge v_2 = \mathbf{S0}) \vee (\neg \varphi(v_1) \wedge v_2 = \mathbf{0})$$
>
> represents (the graph of) K_R, for the same reason as in the last paragraph. (Actually this formula even functionally represents K_R, as the reader can verify.) ⊣

2. If R is a representable binary relation and f, g are representable functions, then

$$\{\vec{a} \mid \langle f(\vec{a}), g(\vec{a})\rangle \in R\}$$

is representable. Similarly for an m-ary relation R and functions $f_1, \ldots, f_m$.

> PROOF.   Its characteristic function at $\vec{a}$ has the value $K_R(f(\vec{a}), g(\vec{a}))$. Thus it is obtained from representable functions by composition. ⊣

For example, suppose that R is a representable ternary relation. Then

$$\{\langle x, y\rangle \mid \langle y, x, x\rangle \in R\}$$

is representable, being

$$\{\langle x, y\rangle \mid \langle I_2^2(x, y), I_1^2(x, y), I_1^2(x, y)\rangle \in R\}.$$

In this way we can rearrange and repeat variables in describing a representable relation.

3. If R is a representable binary relation, then so is

$$P = \{\langle a, b\rangle \mid \text{for some } c \leq b, \langle a, c\rangle \in R\}.$$

PROOF.   We have from catalog item 0 that if

$$Q = \{\langle a, b\rangle \mid \text{for some } c < b, \langle a, c\rangle \in R\},$$

then Q is representable. And

$$\langle a, b\rangle \in P \Leftrightarrow \langle a, S(b)\rangle \in Q$$
$$\Leftrightarrow \langle I_1^2(a, b), S(I_2^2(a, b))\rangle \in Q.$$

Hence by catalog item 2, P is representable.   ⊣

More generally, if R is a representable $(m + 1)$-ary relation, then

$$\{\langle a_1, \ldots, a_m, b\rangle \mid \text{for some } c \leq b, \langle a_1, \ldots, a_m, c\rangle \in R\}$$

is also representable. In vector notation this relation becomes

$$\{\langle \vec{a}, b\rangle \mid \text{for some } c \leq b, \langle \vec{a}, c\rangle \in R\}.$$

Similarly
$$\{\langle \vec{a}, b\rangle \mid \text{for all } c \leq b, \langle \vec{a}, c\rangle \in R\}$$

is representable.

4. The divisibility relation

$$\{\langle a, b\rangle \mid a \text{ divides } b \text{ in } \mathbb{N}\}$$

is representable.

PROOF.   We have a divides b iff for some $q \leq b, a \cdot q = b$. We know that $\{\langle a, b, q\rangle \mid a \cdot q = b\}$ is representable, as it has a quantifier-free definition. Upon applying the above items, we get the divisibility relation. (In yet further detail, from catalog item 3 we get the representability of

$$R = \{\langle a, b, c\rangle \mid \text{for some } q \leq c, \ a \cdot q = b\}$$

and a divides b iff $\langle a, b, b\rangle \in R$.)   ⊣

5. The set of primes is representable.
6. The set of pairs of adjacent primes is representable.

PROOF. $\langle a, b\rangle$ is a pair of adjacent primes iff a is prime and b is prime and $a < b$ and there does not exist any $c < b$ such that

$a < c$ and c is prime. The right side of this equivalence is easily formalized by a numeralwise determined formula. $\dashv$

Note (for future use in Section 3.8) that we have not yet used the fact that exponentiation is representable.

Observe that as this catalog progresses, we are in effect building up a "language" $\mathcal{L}$ such that anything (any relation, any function) that is $\mathcal{L}$-definable (in $\mathfrak{N}$) will be certain to be representable in our theory. Thus, Theorem 33I tells us that (a) atomic formulas are allowed in $\mathcal{L}$, (b) all sentential connectives are permitted, and (c) bounded quantifiers can be used. (Unbounded quantifiers are not in general allowed.) Then our catalog gradually adds particular predicate symbols and function symbols; catalog item 6 adds a two-place predicate symbol for "adjacent primality"; and catalog item 7 will add a function symbol for the prime-listing function. Theorem 33L justifies using these function symbols inside expressions of $\mathcal{L}$.

7. The function whose value at a is p_a, the $(a + 1)$st prime, is representable. (Thus $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7$, $p_4 = 11$, and so forth.)

PROOF. $p_a = b$ iff b is prime and there exists some $c \leq b^{a^2}$, such that (i)–(iii) hold:

(i) 2 does not divide c.

(ii) For any $q < b$ and any $r \leq b$, if $\langle q, r \rangle$ is a pair of adjacent primes, then for all $j < c$,

$$q^j \text{ divides } c \iff r^{j+1} \text{ divides } c.$$

(iii) b^a divides c and b^{a+1} does not.

This equivalence is not obvious, but at least the relation defined by the right-hand side is representable. To verify the equivalence, first note that if $p_a = b$, then we can take

$$c = 2^0 \cdot 3^1 \cdot 5^2 \cdot \ldots \cdot p_a^a.$$

It is easy to check that this value of c meets all the conditions. Conversely, suppose c is a number meeting conditions (i)–(iii). We claim that c must be

$$2^0 \cdot 3^1 \cdot \ldots \cdot b^a \cdot \text{powers of larger primes.}$$

Certainly the exponent of 2 in c is 0, by (i). We can use (ii) to work our way across to the prime b. But by (iii) the exponent of b is a, so b must be the $(a + 1)$st prime, p_a. $\dashv$

This function will be very useful in encoding finite sequences of numbers into single numbers. Let

$$\langle a_0, \ldots, a_m \rangle = p_0^{a_0+1} \cdot \cdots \cdot p_m^{a_m+1}$$
$$= \prod_{i \le m} p_i^{a_i+1}.$$

This holds also for $m = -1$; we define $\langle\ \rangle = 1$. For example,

$$\langle 2, 1 \rangle = 2^3 \cdot 3^2 = 72.$$

The idea is that 72 safely encodes the pair $\langle 2, 1 \rangle$.

There are other ways to encode pairs of numbers and finite sequences of numbers. In Section 3.8, we will make use of a pairing function

$$J(a, b) = \frac{1}{2}[(a + b)^2 + 3a + b]$$

that has the advantage of growing at a polynomial rate, unlike the growth rate of $2^{a+1}3^{b+1}$. Here is a very different way to encode, for example, the numbers 24, 117, 11 (in that order). First we convert to numerals in base 9: 26, 140, 12. Secondly, we concatenate these numerals, separated by 9's: 269140912. The triple is encoded by the number thereby designated (in base 10), that is, 269,140,912. This method may seem tricky, but it produces a result that is *much* smaller than $2^{25}3^{118}5^{12}$, which requires 73 digits in base 10.

8. For each m, the function whose value at $a_0, \ldots, a_m$ is $\langle a_0, \ldots, a_m \rangle$ is representable.

9. There is a representable function (whose value at $\langle a, b \rangle$ is written $(a)_b$) such that for $b \le m$,

$$(\langle a_0, \ldots, a_m \rangle)_b = a_b.$$

(This is our "decoding" function. For example, $(72)_0 = 2$ and $(72)_1 = 1$.)

> PROOF. We define $(a)_b$ to be the least n such that either $a = 0$ or p_b^{n+2} does not divide a. (There always *is* such an n.) Observe that $(0)_b = 0$, and for $a \neq 0$, $(a)_b$ is one less than the exponent of p_b in the prime factorization of a (but not less than 0). Hence for $b \le m$,
>
> $$(\langle a_0, \ldots, a_m \rangle)_b = a_b.$$
>
> To prove representability we use the least-zero operator. Let
>
> $$R = \{\langle a, b, n \rangle \mid a = 0 \text{ or } p_b^{n+2} \text{ does not divide } a\}.$$
>
> Then $(a)_b = \mu n[K_{\overline{R}}(a, b, n) = 0]$, where $\overline{R}$ is the complement of R.　　　　　　　　　　　　　　　　　⊣

Since the method used in the above proof will be useful elsewhere as well, we here state it separately:

THEOREM 33N   Assume that R is a representable relation such that for every $\vec{a}$ there is some n such that $\langle \vec{a}, n \rangle \in R$. Then the function f defined by

$$f(\vec{a}) = \text{the least } n \text{ such that } \langle \vec{a}, n \rangle \in R$$

is representable.

PROOF. $f(\vec{a}) = \mu n[K_{\overline{R}}(\vec{a}, n) = 0]$.   ⊣

We will later use the notation

$$f(\vec{a}) = \mu n[\langle \vec{a}, n \rangle \in R].$$

10. Say that b is a *sequence number* iff for some $m \geq -1$ and some $a_0, \ldots, a_m$,

$$b = \langle a_0, \ldots, a_m \rangle.$$

(When $m = -1$ we get $\langle \ \rangle = 1$.) Then the set of sequence numbers is representable.

PROOF.   Exercise 5.   ⊣

11. There is a representable function lh such that

$$\text{lh}\langle a_0, \ldots, a_m \rangle = m + 1.$$

(Here "lh" stands for "length." For example, $\text{lh } 72 = 2$.)

PROOF.   We define lh a to be the least n such that either $a = 0$ or p_n does not divide a. This works.   ⊣

12. There is a representable function (whose value at $\langle a, b \rangle$ is called the *restriction* of a to b, written $a \upharpoonright b$) such that for any $b \leq m + 1$,

$$\langle a_0, \ldots, a_m \rangle \upharpoonright b = \langle a_0, \ldots, a_{b-1} \rangle.$$

PROOF.   Let $a \upharpoonright b$ be the least n such that either $a = 0$ or both $n \neq 0$ and for any $j < b$, any $k < a$

$$p_j^k \text{ divides } a \ \Rightarrow \ p_j^k \text{ divides } n.$$

This works.   ⊣

13. (Primitive recursion) With a $(k+1)$-place function f we associate another function $\overline{f}$ such that $\overline{f}(a, b_1, \ldots, b_k)$ encodes the values of $f(j, b_1, \ldots, b_k)$ for all $j < a$. Specifically, let

$$\overline{f}(a, \vec{b}) = \langle f(0, \vec{b}), \ldots, f(a-1, \vec{b}) \rangle.$$

For example, $\overline{f}(0, \vec{b}) = \langle \ \rangle = 1$, encoding the first zero values of f. $\overline{f}(1, \vec{b}) = \langle f(0, \vec{b}) \rangle$. In any case, $\overline{f}(a, \vec{b})$ is a sequence number of length a, encoding the first a values of f.

Now suppose we are given a $(k + 2)$-place function g. There exists a unique function f satisfying

$$f(a, \vec{b}) = g(\overline{f}(a, \vec{b}), a, \vec{b}).$$

For example,

$$f(0, \vec{b}) = g(\langle\,\rangle, 0, \vec{b}),$$
$$f(1, \vec{b}) = g(\langle f(0, \vec{b})\rangle, 1, \vec{b}).$$

(The existence and uniqueness of this f should be intuitively clear. For a proof, we can apply the recursion theorem of Section 1.4, obtaining first $\overline{f}$ and then extracting f.)

THEOREM 33P   Let g be a $(k + 2)$-place function and let f be the unique $(k + 1)$-place function such that for all a and $(k$-tuples$)$ $\vec{b}$,

$$f(a, \vec{b}) = g(\overline{f}(a, \vec{b}), a, \vec{b}).$$

If g is representable, then so is f.

PROOF.   First we claim that $\overline{f}$ is representable.
    This follows from the fact that

$\overline{f}(a, \vec{b}) =$ the least s such that s is a sequence number of
        length a and for i less than a, $(s)_i = g(s \upharpoonright i, i, \vec{b})$.

It then follows that f is representable, since

$$f(a, \vec{b}) = g(\overline{f}(a, \vec{b}), a, \vec{b})$$

and the functions on the right are representable.   ⊣

Actually the phrase "primitive recursion" is more commonly applied to a simpler version of this, given in Exercise 8.

14. For a representable function F, the function whose value at $a, \vec{b}$ is

$$\prod_{i<a} F(i, \vec{b})$$

is also representable. Similarly with Σ in place of Π. (For $a = 0$, we use the standard conventions: The empty product — the product of no numbers — is 1, and the empty sum is 0.)

PROOF.   Call this function G; then

$$G(0, \vec{b}) = 1,$$
$$G(a + 1, \vec{b}) = F(a, \vec{b}) \cdot G(a, \vec{b}).$$

Apply Exercise 8.   ⊣

15. Define the *concatenation* of a and b, $a * b$, by

$$a * b = a \cdot \prod_{i<\text{lh}\,b} p_{i+\text{lh}\,a}^{(b)_i + 1}.$$

This is a representable function of a and b, and

$$\langle a_1, \ldots, a_m \rangle * \langle b_1, \ldots, b_n \rangle = \langle a_1, \ldots, a_m, b_1, \ldots, b_n \rangle.$$

The concatenation operation has the further property of being associative on sequence numbers.

16. We will also want a "capital asterisk" operation. Let

$$*_{i<a} f(i) = f(0) * f(1) * \cdots * f(a-1).$$

For a representable function F, the function whose value at a, $\vec{b}$ is $*_{i<a} F(i, \vec{b})$ is representable.

PROOF. $*_{i<0} F(i, \vec{b}) = \langle\,\rangle = 1$ and

$$*_{i<a+1} F(i, \vec{b}) = *_{i<a} F(i, \vec{b}) * F(a, \vec{b}).$$

So this is just like catalog item 14. $\dashv$

Exercises

1. Show that in the structure $(\mathbb{N}; \cdot, E)$ we can define the addition relation $\{\langle m, n, m+n\rangle \mid m, n \text{ in } \mathbb{N}\}$. Conclude that in this structure $\{0\}$, the ordering relation $<$, and the successor relation $\{\langle n, S(n)\rangle \mid n \in \mathbb{N}\}$ are definable. (*Remark*: This result can be strengthened by replacing the structure $(\mathbb{N}; \cdot, E)$ by simply $(\mathbb{N}; E)$. The multiplication relation is definable here, by exploiting one of the laws of exponents: $(d^a)^b = d^{ab}$.)

2. Prove Theorem 33C, stating that true (in $\mathfrak{N}$) quantifier-free sentences are theorems of A_E. (See the outline given there.)

3. A theory T (in a language with $\mathbf{0}$ and $\mathbf{S}$) is called ω-*complete* iff for any formula φ and variable x, if $\varphi_{\mathbf{S}^n \mathbf{0}}^x$ belongs to T for every natural number n, then $\forall x \varphi$ belongs to T. Show that if T is a consistent ω-complete theory in the language of $\mathfrak{N}$ and if $A_E \subseteq T$, then $T = \mathrm{Th}\,\mathfrak{N}$.

4. Show that in the proof preceding Theorem 33L, formula (4) is logically implied by the set consisting of formulas (1), (2), and (3).

5. Show that the set of sequence numbers is representable (catalog item 10).

6. Is 3 a sequence number? What is lh 3? Find $(1*3)*6$ and $1*(3*6)$.

7. Establish the following facts:
   (a) $a + 1 < p_a$.
   (b) $(b)_k \leq b$; equality holds iff $b = 0$.
   (c) lh $a \leq a$; equality holds iff $a = 0$.
   (d) $a \restriction i \leq a$.
   (e) $\mathrm{lh}(a \restriction i)$ is the smaller of i and lh a.

8. Let g and h be representable functions, and assume that

$$f(0, b) = g(b),$$
$$f(a + 1, b) = h(f(a, b), a, b).$$

Show that f is representable.

9. Show that there is a representable function f such that for every $n, a_0, \ldots, a_n$,

$$f(\langle a_0, \ldots, a_n \rangle) = a_n.$$

(For example, $f(72) = 1$ and $f(750) = 2$.)

10. Assume that R is a representable relation and that g and h are representable functions. Show that f is representable, where

$$f(\vec{a}) = \begin{cases} g(\vec{a}) & \text{if } \vec{a} \in R, \\ h(\vec{a}) & \text{if } \vec{a} \notin R. \end{cases}$$

11. (Monotone recursion) Assume that R is a representable binary relation on $\mathbb{N}$. Let C be the smallest subset of $\mathbb{N}$ (i.e., the intersection of all subsets) such that for all $n, a_0, \ldots, a_{n-1}, b$,

$$\langle \langle a_0, \ldots, a_{n-1} \rangle, b \rangle \in R \ \& \ a_i \in C \ (\text{for all } i < n) \ \Rightarrow \ b \in C.$$

Further assume that (1) for all $n, a_0, \ldots, a_{n-1}, b$,

$$\langle \langle a_0, \ldots, a_{n-1} \rangle, b \rangle \in R \ \Rightarrow \ a_i < b \ (\text{for all } i < n),$$

and (2) there is a representable function f such that for all n, $a_0, \ldots, a_{n-1}, b$,

$$\langle \langle a_0, \ldots, a_{n-1} \rangle, b \rangle \in R \ \Rightarrow \ n < f(b)$$

Show that C is representable. (C is, in a sense, generated by R. $C \neq \varnothing$ in general because if $\langle \langle \ \rangle, b \rangle \in R$, then $b \in C$.)

SECTION 3.4
Arithmetization of Syntax

In this section we intend to develop two themes:

1. Certain assertions about wffs can be converted into assertions about natural numbers (by assigning numbers to expressions).

2. These (English) assertions about natural numbers can in many cases be translated into the formal language. And the theory $\text{Cn} \, A_E$ is strong enough to prove many of the translations so obtained.

This will give us the ability to construct formulas that, by expressing facts about numbers, indirectly express facts about formulas (even about themselves!). Such an ability will be exploited in Section 3.5 to obtain results of undefinability and undecidability.

Gödel Numbers

We first want to assign numbers to expressions of the formal language. Recall that the symbols of our language are those listed in Table IX.

TABLE IX

Parameters	Logical symbols
0. $\forall$	1. (
2. **0**	3.)
4. **S**	5. $\neg$
6. $<$	7. $\rightarrow$
8. $+$	9. $=$
10. $\cdot$	11. v_1
12. **E**	13. v_2, etc.

There is a function h assigning to each symbol the integer listed to its left. Thus $h(\forall) = 0$, $h(\mathbf{0}) = 2$, and $h(v_i) = 9 + 2i$. In order to make our subsequent work more widely applicable, we will assume only that we have some language with **0** and **S** which is *recursively numbered*. By this we mean that we have a one-to-one function h from the parameters of that language into the even numbers such that the two relations

$$\{\langle k, m \rangle \mid k \text{ is the value of } h \text{ at some } m\text{-place predicate parameter}\}$$

and

$$\{\langle k, m \rangle \mid k \text{ is the value of } h \text{ at some } m\text{-place function symbol}\}$$

are both representable in Cn A_E. Of course in the case of the language of $\mathfrak{N}$ these sets are even finite. The first set is $\{\langle 6, 2 \rangle\}$ and the second is

$$\{\langle 2, 0 \rangle, \langle 4, 1 \rangle, \langle 8, 2 \rangle, \langle 10, 2 \rangle, \langle 12, 2 \rangle\}.$$

We define h on the logical symbols as before; thus $h(s)$ is an odd number for each logical symbol s.

For an expression $\varepsilon = s_0 \cdots s_n$ of the language we define its Gödel number, $\sharp(\varepsilon)$, by

$$\sharp(s_0 \cdots s_n) = \langle h(s_0), \ldots, h(s_n) \rangle.$$

For example, using our original function h for the language of $\mathfrak{N}$, we obtain

$$\sharp(\exists\, v_3\ v_3 = \mathbf{0})$$
$$= \sharp((\neg \forall\, v_3(\neg = v_3\mathbf{0})))$$
$$= \langle 1, 5, 0, 15, 1, 5, 9, 15, 2, 3, 3 \rangle$$
$$= 2^2 \cdot 3^6 \cdot 5^1 \cdot 7^{16} \cdot 11^2 \cdot 13^6 \cdot 17^{10} \cdot 19^{16} \cdot 23^3 \cdot 29^4 \cdot 31^4.$$

This is a large number, being of the order of 1.3×10^{75}. To a set Φ of expressions we assign the set

$$\sharp\Phi = \{\sharp(\varepsilon) \mid \varepsilon \in \Phi\}$$

of Gödel numbers.

To a sequence $\langle \alpha_0, \ldots, \alpha_n \rangle$ of expressions (such as a deduction), we assign the number

$$\mathcal{G}(\langle \alpha_0, \ldots, \alpha_n \rangle) = \langle \sharp\alpha_0, \ldots, \sharp\alpha_n \rangle.$$

We now proceed to show that various relations and functions having to do with Gödel numbers are representable in Cn A_E (and hence are recursive). As in the preceding section, whenever we say that a relation or function is representable (without specifying a theory) we mean that it is representable in the theory Cn A_E.

We will make use of certain abbreviations in the language we use (i.e., English, although it is coming to differ more and more from what one ordinarily thinks of as English). For "there is a number a such that" we write "$\exists a$." In the same spirit, "$\exists a, b < c$" means "there are numbers a and b both of which are less than c such that." Similarly, we may employ "$\forall$." We would not have dared to employ such abbreviations in Chapter 2, for fear of creating confusion between the formal language and the meta-language (English). But by now we trust the reader to avoid such erroneous ways.

1. The set of Gödel numbers of variables is representable.

PROOF.   It is $\{a \mid (\exists b < a)a = \langle 11 + 2b \rangle\}$. It follows from results of the preceding section that this is a representable set. $\dashv$

2. The set of Gödel numbers of terms is representable.

PROOF.   The set of terms was defined inductively. And terms were built up from constituents with smaller Gödel numbers. We will treat this case in some detail, since it is typical of the argument used for inductively defined relations.

Let f be the characteristic function of the set of Gödel numbers of terms. From the definition of "term" we obtain

$$f(a) = \begin{cases} 1 & \text{if } a \text{ is the Gödel number of a variable,} \\ 1 & \text{if } (\exists i < \square, \exists k < a) \, [i \text{ is a sequence number} \\ & \quad \& \, (\forall j < \text{lh } i)f((i)_j) = 1 \, \& \, k \text{ is the value of} \\ & \quad h \text{ at some } (\text{lh } i)\text{-place function symbol } \& \\ & \quad a = \langle k \rangle * *_{j < \text{lh } i}(i)_j], \\ 0 & \text{otherwise.} \end{cases}$$

But what upper bound for i can we use in place of that "$\square$" symbol? Before we can argue that f is representable, we will

need an upper bound on i that depends in some representable way on a.

The claim is that we can take $i < a^{a \operatorname{lh} a}$. To see this, suppose that $a = \sharp s t_1 \cdots t_n$ (where s is an n-place function symbol and $t_1, \ldots, t_n$ are terms). Then we want to take $i = \langle \sharp t_1, \ldots, \sharp t_n \rangle$. How big could this be, in terms of a? We have the bounds:

$$
\begin{aligned}
i &= 2^{\sharp t_1 + 1} \cdots p_{n-1}^{\sharp t_n + 1} \\
&\leq 2^a \cdots p_{n-1}^a \\
&< 2^a \cdots p_{\operatorname{lh} a - 1}^a \quad \text{because } n = \operatorname{lh} i < \operatorname{lh} a \\
&\leq a^a \cdots a^a \quad (\operatorname{lh} a \text{ times}) \quad \text{because } a = 2^{(a)_0 + 1} \cdots p_{\operatorname{lh} a - 1}^{(a)_{\operatorname{lh} a - 1} + 1} \geq p_{\operatorname{lh} a - 1} \\
&= (a^a)^{\operatorname{lh} a} = a^{a \operatorname{lh} a}
\end{aligned}
$$

So in the above equation for f, we replace $\square$ by $a^{a \operatorname{lh} a}$.

Although the right side of this equation refers to f, it refers only to $f((i)_j)$, where $(i)_j < a$. This feature permits us to apply primitive recursion. $f(a) = g(\overline{f}(a), a)$, where

$$
g(s, a) = \begin{cases}
1 & \text{if } a \text{ is the Gödel number of a variable,} \\
1 & \text{if } (\exists i < a^{a \operatorname{lh} a}, \exists k < a) \, [i \text{ is a sequence number} \\
& \quad \& \, (\forall j < \operatorname{lh} i)(s)_{(i)_j} = 1 \, \& \, k \text{ is the value of} \\
& \quad h \text{ at some } (\operatorname{lh} i)\text{-place function symbol } \& \\
& \quad a = \langle k \rangle * *_{j < \operatorname{lh} i}(i)_j], \\
0 & \text{otherwise.}
\end{cases}
$$

For if in this equation we set s equal to $\overline{f}(a)$, then $(s)_{(i)_j} = f((i)_j)$ for $(i)_j < a$. Hence by Theorem 33P, f is representable provided that g is.

It remains to show that g is representable. But this is straightforward, by using results of the preceding section. Briefly, the graph of g is the union of three relations, corresponding to the three clauses in the above equation. Each of the three is obtained from equality and other representable relations by bounded quantification and the substitution of representable functions.   ⊣

3. The set of Gödel numbers of atomic formulas is representable.

PROOF. a is the Gödel number of an atomic formula iff $(\exists i < a^{a \operatorname{lh} a}, \exists k < a) \, [i$ is a sequence number $\& \, (\forall j < \operatorname{lh} i)(i)_j$ is the Gödel number of a term $\& \, k$ is the value of h at some $(\operatorname{lh} i)$-place predicate symbol $\& \, a = \langle k \rangle * *_{j < \operatorname{lh} i}(i)_j]$.   ⊣

4. The set of Gödel numbers of wffs is representable.

PROOF. The wffs were inductively defined. Let f be the characteristic function of the set, then

$$
f(a) = \begin{cases}
1 & \text{if } a \text{ is the Gödel number of an atomic formula,} \\
1 & \text{if } (\exists i < a)[a = \langle h(()), h(\neg) \rangle * i * \langle h()) \rangle \\
  & \quad \& \ f(i) = 1], \\
1 & \text{if } (\exists i, j < a)[a = \langle h(()) \rangle * i * \langle h(\rightarrow) \rangle * j * \langle h()) \rangle \\
  & \quad \& \ f(i) = f(j) = 1], \\
1 & \text{if } (\exists i, j < a)[a = \langle h(\forall) \rangle * i * j \ \& \ i \text{ is the Gödel} \\
  & \quad \text{number of a variable and } f(j) = 1], \\
0 & \text{otherwise.}
\end{cases}
$$

By the same argument used for the set of Gödel numbers of terms, we have the representability of f. ⊣

5. There is a representable function Sb such that for a term or formula α, variable x, and term t,

$$
\text{Sb}(\sharp\alpha, \sharp x, \sharp t) = \sharp\alpha_t^x.
$$

PROOF. We will need to define $\text{Sb}(a, b, c)$ making use of values $\text{Sb}(i, b, c)$ where $i < a$. As in the case of catalog item 2 (the characteristic function of the set of terms), it will then be possible to show that both $\overline{\text{Sb}}$ and Sb are representable.

The function Sb is described by the following six clauses (i)–(vi):

(i) If a is the Gödel number of a variable and $a = b$ then

$$
\text{Sb}(a, b, c) = c.
$$

(ii) If $(\exists i < a^{a \, \text{lh} \, a}, \exists k < a)[i$ is a sequence number $\& \ (\forall j < \text{lh} \, i)(i)_j$ is the Gödel number of a term $\& \ k$ is the value of h at some (lh i)-place function or predicate symbol $\& \ a = \langle k \rangle * *_{j < \text{lh} \, i}(i)_j]$ then

$$
\text{Sb}(a, b, c) = \langle k \rangle * *_{j < \text{lh} \, i} \text{Sb}((i)_j, b, c)
$$

for that i and k.

(iii) If $(\exists i < a)[i$ is the Gödel number of a wff $\& \ a = \langle h(()), h(\neg) \rangle * i * \langle h()) \rangle]$ then

$$
\text{Sb}(a, b, c) = \langle h(()), h(\neg) \rangle * \text{Sb}(i, b, c) * \langle h()) \rangle
$$

for that i.

(iv) If $(\exists i, j < a)[i$ and j are Gödel numbers of wffs $\& \ a = \langle h(()) \rangle * i * \langle h(\rightarrow) \rangle * j * \langle h()) \rangle]$ then

$$
\text{Sb}(a, b, c) = \langle h(()) \rangle * \text{Sb}(i, b, c) * \langle h(\rightarrow) \rangle * \text{Sb}(j, b, c) * \langle h()) \rangle
$$

for that i and j.

(v) If $(\exists i, j < a)[i$ is the Gödel number of a variable & $i \neq b$ & j is the Gödel number of a wff & $a = \langle h(\forall) \rangle * i * j]$ then

$$\mathrm{Sb}(a, b, c) = \langle h(\forall) \rangle * i * \mathrm{Sb}(j, b, c)$$

for that i and j.

(vi) If none of the above conditions on a and b are met (where we ignore the displayed equation for $\mathrm{Sb}(a, b, c)$) then

$$\mathrm{Sb}(a, b, c) = a.$$

Then the function Sb is obtained by primitive recursion

$$\mathrm{Sb}(a, b, c) = G(\overline{\mathrm{Sb}}(a, b, c), a, b, c)$$

where G is a 4-place function. The graph of G is the union of six 5-ary relations

$$G = R_1 \cup R_2 \cup R_3 \cup R_4 \cup R_5 \cup R_6$$

corresponding to the six clauses above.

The first of the six is

$R_1 = \{\langle s, a, b, c, d \rangle \mid a$ is the Gödel number of a variable & $a = b$ & $d = c\}$.

The second one is

$R_2 = \{\langle s, a, b, c, d \rangle \mid (\exists i < a^{a \, \mathrm{lh} \, a}, \exists k < a)[i$ is a sequence number & $(\forall j < \mathrm{lh}\, i)(i)_j$ is the Gödel number of a term & k is the value of h at some $(\mathrm{lh}\, i)$-place function or predicate symbol & $a = \langle k \rangle * *_{j < \mathrm{lh}\, i}(i)_j$ & $d = \langle k \rangle * *_{j < \mathrm{lh}\, i}(s)_{(i)_j}]\}$

and the others are similar translations of the corresponding clauses in the description of Sb.

It is necessary to note that G is indeed a function; it is single-valued. This is because no two clauses could apply to one number a. And if, for example, clause (ii) applies to a, then we know from Section 2.3 that the numbers i and k are uniquely determined.

Finally, we apply the usual methods to verify that R_1–R_6 are representable, so G is representable, so $\overline{\mathrm{Sb}}$ is representable, so Sb is representable. (Substitution is a complicated operation!)   ⊣

6. The function whose value at n is $\sharp(\mathbf{S}^n \mathbf{0})$ is representable.

PROOF.   Call this function f; then

$$f(0) = \langle h(\mathbf{0}) \rangle,$$
$$f(n + 1) = \langle h(\mathbf{S}) \rangle * f(n).$$

Apply Exercise 8 of the preceding section.   ⊣

7. There is a representable relation Fr such that for a term or formula α and a variable x,

$$\langle \sharp \alpha, \sharp x \rangle \in \mathrm{Fr} \Leftrightarrow x \text{ occurs free in } \alpha.$$

PROOF. $\langle a, b \rangle \in$ Fr $\Leftrightarrow$ Sb$(a, b, \sharp \mathbf{0}) \neq a$. $\dashv$

8. The set of Gödel numbers of sentences is representable.

PROOF. a is the Gödel number of a sentence iff a is the Gödel number of a formula and for any $b < a$, if b is the Gödel number of a variable then $\langle a, b \rangle \notin$ Fr. $\dashv$

9. There is a representable relation Sbl such that for a formula α, variable x, and term t, $\langle \sharp a, \sharp x, \sharp t \rangle \in$ Sbl iff t is substitutable for x in α.

PROOF.   Exercise 1. $\dashv$

10. The relation Gen, where $\langle a, b \rangle \in$ Gen iff a is the Gödel number of a formula and b is the Gödel number of a generalization of that formula, is representable.

PROOF. $\langle a, b \rangle \in$ Gen iff $a = b$ or $(\exists i, j < b)[i$ is the Gödel number of a variable and $\langle a, j \rangle \in$ Gen and $b = \langle (h(\forall)) * i * j]$. Apply the usual argument to the characteristic function of Gen. $\dashv$

11. The set of Gödel numbers of tautologies is representable.

The set of tautologies is informally decidable since we can use the method of truth tables. To obtain representability, we recast truth tables in terms of Gödel numbers. There are several preliminary steps:

11.1 The relation R, such that $\langle a, b \rangle \in R$ iff a is the Gödel number of a formula α and b is the Gödel number of a prime constituent of α, is representable.

PROOF. $\langle a, b \rangle \in R \Leftrightarrow a$ is the Gödel number of a formula and one of the following:

(i) $a = b$ & $(a)_0 \neq h(()$.
(ii) $(\exists i < a)[a = \langle h(()), h(\neg) \rangle * i * \langle (h()) \rangle$ and $\langle i, b \rangle \in R]$.
(iii) The analogue to (ii) but with $\rightarrow$.

Apply the usual argument to the characteristic function of R. $\dashv$

11.2 There is a representable function P such that for a formula α, $P(\sharp \alpha) = \langle \sharp \beta_1, \ldots, \sharp \beta_n \rangle$, the list of Gödel numbers of prime constituents of α, in numerical order.

PROOF.   First define a function g for locating the next prime constituent in $\natural a$ after $\natural y$ (where $\natural a$ is the formula α for which $a = \sharp \alpha$).

$g(a, y) =$ the least n such that either $n = a + 1$ or both $y < n$ and $\langle a, n \rangle \in R$.

Next define a function h such that $h(a, n)$ gives the $(n+1)$st prime constituent of $\natural a$ (if there are that many):

$h(a, 0) = g(a, 0)$ $h(a, n + 1) = g(a, h(a, n))$.

Finally, let $P(a) = *_{i<k}\langle h(a, i)\rangle$ where k is the least number for which $h(a, k) > a$.   ⊣

11.3 Say that the integer v *encodes a truth assignment* for α iff v is a sequence number and lh $v = $ lh $P(\sharp\alpha)$ and $(\forall i < $ lh $v)(\exists e < 2)(v)_i = \langle (P(\sharp\alpha))_i, e\rangle$. This is a representable condition on v and $\sharp\alpha$.

For example, if $P(\sharp\alpha) = \langle \sharp\beta_0, \ldots, \sharp\beta_n\rangle$, then

$$v = \langle \langle \sharp\beta_0, e_0\rangle, \ldots, \langle \sharp\beta_n, e_n\rangle\rangle,$$

where each e_i is 0 or 1. We will later need an upper bound for v in terms of $\sharp\alpha$. The largest v is obtained when each e_i is 1. Also $\sharp\beta_i \leq \sharp\alpha$, so that

$$v \leq \langle \langle \sharp\alpha, 1\rangle, \ldots, \langle \sharp\alpha, 1\rangle\rangle$$
$$= *_{i < \text{lh } P(\sharp\alpha)}\langle \langle \sharp\alpha, 1\rangle\rangle.$$

11.4 There is a representable relation Tr such that for a formula α and a v which encodes a truth assignment for α (or more), $\langle \sharp\alpha, v\rangle \in$ Tr iff that truth assignment satisfies α.

PROOF.   Exercise 2.   ⊣

Finally, α is a tautology iff α is a formula and for every v encoding a truth assignment for α, $\langle \sharp\alpha, v\rangle \in$ Tr. The (English) quantifier on v can be bounded by a representable function of $\sharp\alpha$, as explained in 11.3.

12. The set of Gödel numbers of formulas of the form $\forall x\, \varphi \rightarrow \varphi_t^x$, where t is a term substitutable for the variable x in φ, is representable.

PROOF. α is of this form iff $(\exists$ wff $\varphi < \alpha)(\exists$ variable $x < \alpha)(\exists$ term $t < \alpha)[t$ is substitutable for x in φ and $\alpha = \forall x\, \varphi \rightarrow \varphi_t^x]$. Here "$\varphi < \alpha$" means that $\sharp\varphi < \sharp\alpha$. This is easily rewritten in terms of Gödel numbers: a belongs to the set iff $(\exists f < a)(\exists x < a)(\exists t < a)$ $[f$ is the Gödel number of a formula & x is the Gödel number of a variable & t is the Gödel number of term and $\langle f, x, t\rangle \in$ Sbl &

$a = \langle h(\,(\,), h(\forall)\rangle * x * f * \langle h(\rightarrow)\rangle * \text{Sb}(f, x, t) * \langle h()\rangle)].$

13. The set of Gödel numbers of formulas of the form $\forall x(\alpha \rightarrow \beta) \rightarrow \forall x\, \alpha \rightarrow \forall x\, \beta$ is representable.

PROOF. γ is of this form iff $(\exists$ variable $x < \gamma)(\exists$ formulas $\alpha, \beta < \gamma)$ $[\gamma = \forall x(\alpha \rightarrow \beta) \rightarrow \forall x\, \alpha \rightarrow \forall x\, \beta]$. This is easily rewritten in terms of Gödel numbers, as in 12.   ⊣

14. The set of Gödel numbers of formulas of the form $\alpha \rightarrow \forall x\, \alpha$, where x does not occur free in α, is representable.

PROOF.   Similar to 13.   ⊣

15. The set of Gödel numbers of formulas of the form $x = x$ is representable.

PROOF.   Similar to 13.   ⊣

16. The set of Gödel numbers of formulas of the form $x = y \rightarrow \alpha \rightarrow \alpha'$, where α is atomic and α' is obtained from α by replacing x at zero or more places by y, is representable.

PROOF.   This is similar to 13, except for the relation of "partial substitution." Let $\langle a, b, x, y \rangle \in$ Psb iff x and y are Gödel numbers of variables, a is the Gödel number of an atomic formula, b is a sequence number of length lh a, and for all $j <$ lh a, either $(a)_j = (b)_j$ or $(a)_j = x$ and $(b)_j = y$. This relation is representable.   ⊣

17. The set of Gödel numbers of logical axioms is representable.

PROOF. α is a logical axiom iff $\exists \beta \leq \alpha$ such that α is a generalization of β and β is in one of the sets in items 11–16.   ⊣

18. For a finite set A of formulas,

$$\{\mathcal{G}(D) \mid D \text{ is a deduction from } A\}$$

is representable. In fact it is enough here for $\sharp A$ to be representable.

PROOF.   A number d belongs to this set iff d is a sequence number of positive length and for every i less than lh d, either

1. $(d)_i \in \sharp A$,
2. $(d)_i$ is the Gödel number of a logical axiom, or
3. $(\exists j, k < i)[(d)_j = \langle h(() \rangle * (d)_k * \langle h(\rightarrow) \rangle * (d)_i * \langle h()) \rangle].$

This is representable whenever $\sharp A$ is, as is certainly the case for finite A.   ⊣

19. Any recursive relation is representable in Cn A_E.

PROOF.   Recall that the relation R is recursive iff there is *some* finite consistent set A of sentences such that some formula ρ represents R in Cn A. (There is no loss of generality in assuming that the language has only finitely many parameters: those in the finite set A, those in ρ, and **0**, **S**, and ∀.) In the case of a unary relation R, we have that $a \in R$ iff the least D which is a deduction from A of either $\rho(\mathbf{S}^a \mathbf{0})$ or $\neg \rho(\mathbf{S}^a \mathbf{0})$ is, in fact, a deduction of the former.

More formally, $a \in R$ iff the last component of $f(a)$ is $\sharp \rho(\mathbf{S}^a \mathbf{0})$, where

$f(a) =$ the least d such that d is in the set of item 18 and the last component of d is either $\sharp \rho(\mathbf{S}^a \mathbf{0})$ or $\sharp \neg \rho(\mathbf{S}^a \mathbf{0})$.

For this (fixed) ρ, there always is such a d.   ⊣

Since the converse to item 19 is immediate, we have

THEOREM 34A   A relation is recursive iff it is representable in the theory Cn A_E.

Henceforth we will usually use the word "recursive" in preference to "representable."

Corollary 34B   Any recursive relation is definable in $\mathfrak{N}$.

20. Now suppose we have a set A of sentences such that $\sharp A$ is recursive. Then $\sharp \operatorname{Cn} A$ need *not* be recursive (as we will show in the next section). But we do have a way of defining $\operatorname{Cn} A$ from A:

$a \in \sharp \operatorname{Cn} A$ iff $\exists d[d$ is the number of a deduction from A
and the last component of d is a and a is the
Gödel number of a sentence]

The part in square brackets is recursive, by the proof to item 18. But we cannot in general put any bound on the number d. The best we can say is that $\sharp \operatorname{Cn} A$ is the domain of a recursive relation (or, as we will say later, is *recursively enumerable*).

Item 20 will play a key role our subsequent work. In particular, it will later be restated as Theorem 35I.

21. If $\sharp A$ is recursive and $\operatorname{Cn} A$ is a complete theory, then $\sharp \operatorname{Cn} A$ is recursive.

In other words, a complete recursively axiomatizable theory is recursive. This is the analogue to Corollary 25G, which asserts that a complete axiomatizable theory is decidable.
  The proof is essentially unchanged. Let (in the consistent case)

$g(s) =$ the least d such that s is not the Gödel number of a sentence,
or d is in the set of item 18 and the last component of d is
either s or is $\langle h((), h(\neg) \rangle * s * \langle h() \rangle)$.

Thus $g(\sharp \sigma)$ is $\mathcal{G}$ of the least deduction of σ or $(\neg \sigma)$ from A. And $s \in \sharp \operatorname{Cn} A$ iff $s > 0$ and the last component of $g(s)$ is s.   ⊣

At this point we might reconsider the plausibility of Church's thesis. Suppose that the relation R is decidable. Then there is a finite list of explicit instructions (a program) for the decision procedure. The procedure itself will presumably consist of certain atomic steps, which are then performed repeatedly. (The reader familiar with computer programming will know that a short program can still require much time for its execution, but some commands will be utilized over and over.) Any one atomic step is presumably very simple.
  By devices akin to Gödel numbering, we can mirror the decision procedure in the integers. The characteristic function of R can then be put in the form

$K_R(\vec{a}) = U[$the least s such that

(i) $(s)_0$ encodes the input $\vec{a}$;
(ii) for all positive $i < \operatorname{lh} s$, $(s)_i$ is obtained from $(s)_{i-1}$ by perfor-
mance of the applicable atomic step;
(iii) the last component of s describes a terminal situation, at which
the computation is completed],

where U (the upshot function) is some simple function that extracts
from the last component of s the answer (affirmative or negative). The
recursiveness of R is now reduced to the recursiveness of U and of
the relations indicated in (i), (ii), and (iii). In special cases, such as
decision procedures provided by the register machines of Section 3.6,
the recursiveness of these components is easily verified. It seems most
improbable that any decision procedure will ever be regarded as effective
and yet will have components that are nonrecursive. For example, in (ii),
it seems that it ought to be possible to make each atomic step extremely
simple, and in particular to make each one recursive.

Exercises

1. Supply a proof for item 9 of this section.
2. Supply a proof for item 11.4 of this section.
3. Use Exercise 11 of Section 3.3 to give a new proof that the set of
   Gödel numbers of terms is representable (item 2).
4. Let T be a consistent recursively axiomatizable theory (in a recur-
   sively numbered language with $\mathbf{0}$ and $\mathbf{S}$). Show that any relation
   representable in T must be recursive.

SECTION 3.5
Incompleteness and Undecidability

In this section we reap the rewards of our work in Sections 3.3 and 3.4.
We have assigned Gödel numbers to expressions, and we have shown
that certain intuitively decidable relations on $\mathbb{N}$ (related to syntactical
notions about expressions) are representable in $\operatorname{Cn} A_E$.

Throughout this section we assume that the language in question is
the language of $\mathfrak{N}$. (This affects the meaning of "Cn" and "theory.")

FIXED-POINT LEMMA   Given any formula β in which only v_1 occurs
free, we can find a sentence σ such that

$$A_E \vdash [\sigma \leftrightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})].$$

We can think of σ as indirectly saying, "β is true of me." Ac-
tually, of course, σ doesn't say anything; it's just a string of sym-
bols. And even when translated into English according to the intended

structure $\mathfrak{N}$, it then talks of numbers and their successors and products and so forth. It is only by virtue of our having associated numbers with expressions that we can think of σ as referring to a formula, in this case to σ itself.

PROOF.   Let $\theta(v_1, v_2, v_3)$ functionally represent in Cn A_E a function whose value at $\langle \sharp\alpha, n \rangle$ is $\sharp(\alpha(\mathbf{S}^n\mathbf{0}))$. (See items 5 and 6 in Section 3.4.) First consider the formula

$$\forall v_3[\theta(v_1, v_1, v_3) \rightarrow \beta(v_3)]. \tag{1}$$

(We may suppose v_3 is substitutable for v_1 in β. The above formula has only v_1 free. It defines in $\mathfrak{N}$ a set to which $\sharp\alpha$ belongs iff $\sharp(\alpha(\mathbf{S}^{\sharp\alpha}\mathbf{0}))$ is in the set defined by β.) Let q be the Gödel number of (1). Let σ be

$$\forall v_3[\theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3) \rightarrow \beta(v_3)].$$

Thus σ is obtained from (1) by replacing v_1, by $\mathbf{S}^q\mathbf{0}$. Notice that σ does assert (under $\mathfrak{N}$) that $\sharp\sigma$ is in the set defined by β. But we must check that

$$\sigma \leftrightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}) \tag{2}$$

is a consequence of A_E. Because θ functionally represents a function whose value at $\langle q, q \rangle$ is $\sharp\sigma$, we have

$$A_E \vdash \forall v_3[\theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3) \leftrightarrow v_3 = \mathbf{S}^{\sharp\sigma}\mathbf{0}]. \tag{3}$$

We can obtain (2) as follows:
($\rightarrow$) It is clear (by looking at σ) that

$$\sigma \vdash \theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^{\sharp\sigma}) \rightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}).$$

And, by (3),

$$A_E \vdash \theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^{\sharp\sigma}\mathbf{0}).$$

Hence

$$A_E; \sigma \vdash \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}),$$

which gives half of (2).
($\leftarrow$) The sentence in (3) implies

$$\beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}) \rightarrow [\forall v_3(\theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3) \rightarrow \beta(v_3))].$$

But the part in square brackets is just σ. $\dashv$

(Sometimes the notation $\ulcorner\sigma\urcorner$ is used for $\mathbf{S}^{\sharp\sigma}\mathbf{0}$. In this notation, the fixed-point lemma states that $A_E \vdash (\sigma \leftrightarrow \beta(\ulcorner\sigma\urcorner))$.)

Our first application of this lemma does not concern the subtheory Cn A_E, and requires only the weaker fact that

$$\models_{\mathfrak{N}} [\sigma \leftrightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})].$$

TARSKI UNDEFINABILITY THEOREM (1933)   The set $\sharp\,\mathrm{Th}\,\mathfrak{N}$ is not definable in $\mathfrak{N}$.

PROOF.   Consider any formula β (which you suspect *might* define $\sharp\,\mathrm{Th}\,\mathfrak{N}$). By the fixed-point lemma (applied to $\neg\,\beta$) we have a sentence σ such that

$$\models_{\mathfrak{N}} [\sigma \leftrightarrow \neg\,\beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})].$$

(If β did define $\sharp\,\mathrm{Th}\,\mathfrak{N}$, then σ would indirectly say "I am false.") Then

$$\models_{\mathfrak{N}} \sigma \iff \not\models_{\mathfrak{N}} \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}),$$

so either σ is true but (its Gödel number is) not in the set β defines, or it is false and in that set. Either way σ shows that β cannot define $\sharp\,\mathrm{Th}\,\mathfrak{N}$.   ⊣

The above theorem gives at once the undecidability of the theory of $\mathfrak{N}$:

COROLLARY 35A $\sharp\,\mathrm{Th}\,\mathfrak{N}$ is not recursive.

PROOF.   Any recursive set is (by Corollary 34B) definable in $\mathfrak{N}$.

                                                                                        ⊣

GÖDEL INCOMPLETENESS THEOREM (1931)   If $A \subseteq \mathrm{Th}\,\mathfrak{N}$ and $\sharp A$ is recursive, then $\mathrm{Cn}\,A$ is not a complete theory.

Thus there is no complete recursive axiomatization of $\mathrm{Th}\,\mathfrak{N}$.

PROOF.   Since $A \subseteq \mathrm{Th}\,\mathfrak{N}$, we have $\mathrm{Cn}\,A \subseteq \mathrm{Th}\,\mathfrak{N}$. If $\mathrm{Cn}\,A$ is a complete theory, then equality holds. But if $\mathrm{Cn}\,A$ is a complete theory, then $\sharp\,\mathrm{Cn}\,A$ is recursive (item 21 of the preceding section). And by the above corollary, $\sharp\,\mathrm{Th}\,\mathfrak{N}$ is not recursive.   ⊣

In particular, $\mathrm{Cn}\,A_E$ is not a complete theory and so is not equal to $\mathrm{Th}\,\mathfrak{N}$. And the incompleteness would not be eliminated by the addition of any recursive set of true axioms. (By a recursive set of sentences we mean of course a set Σ for which $\sharp\Sigma$ is recursive.)

We can extract more information from the proof of Gödel's theorem. Suppose we have a particular recursive $A \subseteq \mathrm{Th}\,\mathfrak{N}$ in mind. Then by item 20 in Section 3.4 we can find a formula β that defines $\sharp\,\mathrm{Cn}\,A$ in $\mathfrak{N}$. The sentence σ produced by the proof of Tarski's theorem is (as we noted there) a true sentence *not* in $\mathrm{Cn}\,A$. This sentence asserts that $\sharp\sigma$ does not belong to the set defined by β, i.e., it indirectly says, "I am not a theorem of A." Thus $A \nvdash \sigma$, and of course $A \nvdash \neg\,\sigma$ as well. This way of viewing the proof is closer to Gödel's original proof, which did not involve a detour through Tarski's theorem. For that matter, Gödel's statement of the theorem did not involve $\mathrm{Th}\,\mathfrak{N}$; we have taken some liberties in the labeling of theorems.

Next we need a lemma which says (roughly) that one can add one new axiom (and hence finitely many new axioms) to a recursive theory without losing the property of recursiveness.

LEMMA 35B   If $\sharp \operatorname{Cn} \Sigma$ is recursive, then $\sharp \operatorname{Cn}(\Sigma; \tau)$ is recursive.

PROOF. $\alpha \in \operatorname{Cn}(\Sigma; \tau) \Leftrightarrow (\tau \rightarrow \alpha) \in \operatorname{Cn} \Sigma$. Thus

$a \in \sharp \operatorname{Cn}(\Sigma; \tau) \iff a$ is the Gödel number of a sentence
$\qquad\qquad$ and $\langle h(() \rangle * \sharp \tau * \langle h(\rightarrow) \rangle * a * \langle h()) \rangle$
$\qquad\qquad$ is in $\sharp \operatorname{Cn} \Sigma$.

This is recursive by the results of the preceding sections.   ⊣

THEOREM 35C (STRONG UNDECIDABILITY OF CN A_E)   Let T be a theory such that $T \cup A_E$ is consistent. Then $\sharp T$ is not recursive.

(Notice that because throughout this section the language in question is the language of $\mathfrak{N}$, the word "theory" here means "theory in the language of $\mathfrak{N}$.")

PROOF.   Let T' be the theory $\operatorname{Cn}(T \cup A_E)$. If $\sharp T$ is recursive, then since A_E is finite we can conclude by the above lemma that $\sharp T'$ is also recursive.

Suppose, then, that $\sharp T'$ is recursive and so is represented in $\operatorname{Cn} A_E$ by some formula β. From the fixed-point lemma we get a sentence σ such that

$$A_E \vdash [\sigma \leftrightarrow \neg \beta(\mathbf{S}^{\sharp \sigma} \mathbf{0})]. \qquad\qquad (*)$$

(Indirectly σ asserts, "I am not in T'.")

$$\begin{aligned}
\sigma \notin T' &\Rightarrow \sharp\sigma \notin \sharp T' \\
&\Rightarrow A_E \vdash \neg \beta(\mathbf{S}^{\sharp\sigma} \mathbf{0}) \\
&\Rightarrow A_E \vdash \sigma \qquad\qquad \text{by } (*) \\
&\Rightarrow \sigma \in T'.
\end{aligned}$$

So we get $\sigma \in T'$. But this, too, is untenable:

$$\begin{aligned}
\sigma \in T' &\Rightarrow \sharp\sigma \in \sharp T' \\
&\Rightarrow A_E \vdash \beta(\mathbf{S}^{\sharp\sigma} \mathbf{0}) \\
&\Rightarrow A_E \vdash \neg\sigma \qquad\qquad \text{by } (*) \\
&\Rightarrow (\neg\sigma) \in T',
\end{aligned}$$

which contradicts the consistency of T'.   ⊣

COROLLARY 35D   Assume that $\sharp \Sigma$ is recursive and $\Sigma \cup A_E$ is consistent. Then $\operatorname{Cn} \Sigma$ is not a complete theory.

PROOF.   A complete recursively axiomatizable theory is recursive (item 21 of Section 3.4). But $\sharp \operatorname{Cn} \Sigma$ is not recursive, by the above theorem.   ⊣

This corollary is Gödel's incompleteness theorem again, but with truth in $\mathfrak{N}$ replaced by consistency with A_E.

CHURCH'S THEOREM (1936)   The set of Gödel numbers of valid sentences (in the language of $\mathfrak{N}$) is not recursive.

PROOF.   In the strong undecidability of Cn A_E, take T to be the smallest theory in the language, the set of valid sentences.   ⊣

The set of Gödel numbers of valid wffs is not recursive either, lest the set of valid sentences be recursive.

This proof applies to the language of $\mathfrak{N}$. For a language with more parameters, the set of valid sentences is still nonrecursive (lest its intersection with the language of $\mathfrak{N}$ be recursive). Actually it is enough for the language to contain at least one two-place predicate symbol. (See Corollary 37G.) On the other hand, *some* lower bound on the language is needed. If the language has $\forall$ as its only parameter (the language of equality), then the set of valid formulas is decidable. (See Exercise 6.) More generally, it is known that if the only parameters are $\forall$ and one-place predicate symbols, then the set of valid formulas is decidable.

Recursive Enumerability

A relation on the natural numbers is said to be *recursively enumerable* iff it is of the form

$$\{\vec{a} \mid \exists b \, \langle \vec{a}, b \rangle \in Q\}$$

with Q recursive. Recursively enumerable relations play an important role in logic. They constitute the formal counterpart to the effectively enumerable relations (as will be explained presently).

(The standard abbreviation for "recursively enumerable" is "r.e." When the term "computable" is used instead of "recursive," then one speaks of *computably enumerable* — abbreviated c.e. — relations.)

Recursively enumerable relations are — like the recursive relations — definable in $\mathfrak{N}$. If $\varphi(v_1, v_2)$ defines in $\mathfrak{N}$ a binary relation Q, then $\exists v_2 \, \varphi(v_1, v_2)$ defines $\{a \mid \exists b \, \langle a, b \rangle \in Q\}$.

THEOREM 35E   The following conditions on an m-ary relation R are equivalent:

1. R is recursively enumerable.
2. R is the domain of some recursive relation Q.
3. For some recursive $(m + 1)$-ary relation Q,

$$R = \{\langle a_1, \ldots, a_m \rangle \mid \exists b \, \langle a_1, \ldots, a_m, b \rangle \in Q\}.$$

4. For some recursive $(m + n)$-ary relation Q,

$$R = \{\langle a_1, \ldots, a_m \rangle \mid \exists b_1, \ldots, b_n \, \langle a_1, \ldots, a_m, b_1, \ldots, b_n \rangle \in Q\}.$$

PROOF.   By definition 1 and 3 are equivalent. Also 2 and 3 are equivalent by our definition (in Chapter 0) of domain and $(m+1)$-tuple.

Clearly 3 implies 4. So we have only to show that 4 implies 3. This is because

$$\exists b_1, \ldots, b_n \, \langle a_1, \ldots, a_m, b_1, \ldots, b_n \rangle \in Q$$
$$\text{iff} \ \exists c \, \langle a_1, \ldots, a_m, (c)_0, \ldots, (c)_{n-1} \rangle \in Q$$

and

$$\{ \langle a_1 \ldots, a_m, c \rangle \mid \langle a_1, \ldots, a_m, (c)_0, \ldots, (c)_{n-1} \rangle \in Q \}$$

is recursive whenever Q is recursive. (Here we have used our sequence decoding function to collapse a string of quantifiers into a single one.)   ⊣

By part 4 of this theorem, R is recursively enumerable iff it is definable in $\mathfrak{N}$ by a formula $\exists x_1 \cdots \exists x_n \varphi$, where φ is numeralwise determined by A_E. In fact, we can require here that φ be quantifier-free; this result was proved in 1961 (with exponentiation) and in 1970 (without exponentiation). The proofs involve some number theory; we will omit them here.

Notice that any recursive relation is also recursively enumerable. For if R is recursive, then it is defined in $\mathfrak{N}$ by a formula $\exists x_1 \cdots \exists x_n \varphi$, where φ is numeralwise determined by A_E and $x_1, \ldots, x_n$ do not occur in φ.

THEOREM 35F   A relation is recursive iff both it and its complement are recursively enumerable.

This is the formal counterpart to the fact (cf. Theorem 17F) that a relation is decidable iff both it and its complement can be effectively enumerated.

PROOF.   If a relation is recursive, then so is its complement, whence both are recursively enumerable.

Conversely, suppose that both P and its complement are recursively enumerable; thus for any $\vec{a}$,

$$\vec{a} \in P \Leftrightarrow \exists b \, \langle \vec{a}, b \rangle \in Q$$
$$\vec{a} \notin P \Leftrightarrow \exists b \, \langle \vec{a}, b \rangle \in R$$

for some recursive Q and R. Let

$$f(\vec{a}) = \text{the least } b \text{ such that either } \langle \vec{a}, b \rangle \in Q \text{ or } \langle \vec{a}, b \rangle \in R.$$

Such a number b always exists, and f is recursive. Finally,

$$\vec{a} \in P \Leftrightarrow \langle \vec{a}, f(\vec{a}) \rangle \in Q,$$

so P is recursive.   ⊣

The recursively enumerable relations constitute the formal counterpart of the effectively enumerable relations. For we have the following

informal result, paralleling a characterization of recursive enumerability given by Theorem 35E.

⋆**LEMMA 35G**   A relation is effectively enumerable iff it is the domain of a decidable relation.

PROOF.   Assume that Q is effectively enumerated by some procedure. Then $\vec{a} \in Q$ iff $\exists n[\vec{a}$ appears in the enumeration in n steps]. The relation defined in square brackets is decidable and has domain Q.

Conversely, to enumerate $\{\langle a, b\rangle \mid \exists n \langle a, b, n\rangle \in R\}$ for decidable R, we check to see if $\langle (m)_0, (m)_1, (m)_2\rangle \in R$ for $m = 0, 1, 2, \ldots$. Whenever the answer is affirmative, we place $\langle (m)_0, (m)_1\rangle$ on the output list.   ⊣

⋆**COROLLARY 35H (CHURCH'S THESIS, SECOND FORM)**   A relation is effectively enumerable iff it is recursively enumerable.

PROOF.   By identifying the class of decidable relations with the class of recursive relations, we automatically identify the domains of decidable relations with domains of recursive relations.   ⊣

The second form of Church's thesis is, in fact, equivalent to the first form. To prove the first form from the second, we use Theorems 35F and 17F.

We have already shown that a recursively axiomatizable theory is recursively enumerable, but using different words. We restate the result here, as it indicates the role recursive enumerability plays in logic.

THEOREM 35I   If A is a set of sentences such that $\sharp A$ is recursive, then $\sharp \operatorname{Cn} A$ is recursively enumerable.

PROOF.   Item 20 of Section 3.4.   ⊣

In particular, $\sharp \operatorname{Cn} A_E$ is recursively enumerable, but (by Theorem 35C) it is not recursive. In the next section, we will see other examples of recursively enumerable sets that are not recursive.

This theorem is the precise counterpart of the informal fact that a theory with a decidable set of axioms is effectively enumerable (Corollaries 25F and 26I). It indicates the gap between what is *provable* in an axiomatic theory and what is *true* in the intended structure. With a recursive set of axioms, all we can possibly obtain is a recursively enumerable set of consequences. But by Tarski's theorem, Th $\mathfrak{N}$ is not even definable in $\mathfrak{N}$, much less recursively enumerable.

Even if we expand the language or add new axioms, the same phenomenon is present. As long as we can recursively distinguish deductions from nondeductions, the set of theorems can be only recursively enumerable. For example, the set of sentences of number theory provable in your favorite system of axiomatic set theory is recursively

enumerable. Furthermore, this set includes A_E and is consistent (unless you have very strange favorites). It follows that this set theory is nonrecursive and incomplete. (This topic is discussed more carefully in Section 3.7.)

Weak Representability

Consider a recursively enumerable set Q, where

$$a \in Q \Leftrightarrow \exists b \, \langle a, b \rangle \in R$$

for recursive R. We know there is a formula ρ that represents R in Cn A_E. Consequently, the formula $\exists v_2 \rho$ defines Q in $\mathfrak{N}$. This formula cannot represent Q in Cn A_E unless Q is recursive. But it can come halfway.

$$
\begin{aligned}
a \in Q &\Rightarrow \langle a, b \rangle \in R & \text{for some } b \\
&\Rightarrow A_E \vdash \rho(\mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0}) & \text{for some } b \\
&\Rightarrow A_E \vdash \exists v_2 \rho(\mathbf{S}^a \mathbf{0}, v_2) \\
a \notin Q &\Rightarrow \langle a, b \rangle \notin R & \text{for all } b \\
&\Rightarrow A_E \vdash \neg \rho(\mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0}) & \text{for all } b \\
&\Rightarrow A_E \nvdash \exists v_2 \rho(\mathbf{S}^a \mathbf{0}, v_2)
\end{aligned}
$$

The last step is justified by the fact that if $A_E \vdash \neg \varphi(\mathbf{S}^b \mathbf{0})$ for all b, then $A_E \nvdash \exists x \, \varphi(x)$. (The term *$\omega$-consistency* is given to this property.) For it is impossible for $\exists x \, \varphi(x)$, $\neg \varphi(\mathbf{S}^0 \mathbf{0})$, $\neg \varphi(\mathbf{S}^1 \mathbf{0})$, . . . all to be true in $\mathfrak{N}$.

Thus we have

$$a \in Q \;\Leftrightarrow\; A_E \vdash \exists v_2 \rho(\mathbf{S}^a \mathbf{0}, v_2).$$

It will be convenient to formulate a definition of this half of representability.

> DEFINITION.   Let Q be an n-ary relation on $\mathbb{N}$, ψ a formula in which only $v_1, \ldots, v_n$ occur free. Then ψ *weakly represents* Q in a theory T iff for every $a_1, \ldots, a_n$ in $\mathfrak{N}$,
>
> $$\langle a_1, \ldots, a_n \rangle \in Q \;\Leftrightarrow\; \psi(\mathbf{S}^{a_1} \mathbf{0}, \ldots, \mathbf{S}^{a_n} \mathbf{0}) \in T.$$

Observe that if Q is representable in a consistent theory T, then Q is also weakly representable in T.

> THEOREM 35J   A relation is weakly representable in Cn A_E iff it is recursively enumerable.

> PROOF.   We just showed that a recursively enumerable unary relation Q is weakly representable in Cn A_E; the same proof applies to n-ary Q with only notational changes. Conversely, let Q be

weakly represented by ψ in Cn A_E. Then

$$\langle a_1, \ldots, a_n \rangle \in Q \Leftrightarrow \exists D \, [D \text{ is a deduction of } \psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0})$$
$$\text{from the axioms } A_E]$$
$$\Leftrightarrow \exists d \, \langle d, f(a_1, \ldots, a_n) \rangle \in P$$

for a certain recursive function f and recursive relation P.   ⊣

Arithmetical Hierarchy

Define a relation on the natural numbers to be *arithmetical* iff it is definable in $\mathfrak{N}$. But some arithmetical relations are, in a sense, more definable than others. We can organize the arithmetical relations into a hierarchy according to how definable the relations are.

Let Σ_1 be the class of recursively enumerable relations; these relations are "one quantifier away" from recursiveness. Extending this idea, we define the class of Σ_k relations and the class of Π_k relations. For example, the first few classes consist of relations of the form shown in the second column:

$$\Sigma_1 : \quad \{\vec{a} \mid \exists b \, \langle \vec{a}, b \rangle \in R\}, \qquad R \text{ recursive.}$$
$$\Pi_1 : \quad \{\vec{a} \mid \forall b \, \langle \vec{a}, b \rangle \in R\}, \qquad R \text{ recursive.}$$
$$\Sigma_2 : \quad \{\vec{a} \mid \exists c \forall b \, \langle \vec{a}, b, c \rangle \in R\}, \quad R \text{ recursive.}$$
$$\Pi_2 : \quad \{\vec{a} \mid \forall c \exists b \, \langle \vec{a}, b, c \rangle \in R\}, \quad R \text{ recursive.}$$

In general, a relation Q is in Π_k iff it is of the form

$$\{\vec{a} \mid \forall b_1 \exists b_2 \cdots \Box b_k \, \langle \vec{a}, \vec{b} \rangle \in R\}$$

for a recursive relation R. Here "$\Box$" is to be replaced by "$\forall$" if k is odd and by "$\exists$" if k is even. Similarly, Q is in Σ_k iff it has the form

$$\{\vec{a} \mid \exists b_1 \forall b_2 \cdots \Box b_k \, \langle \vec{a}, \vec{b} \rangle \in R\}$$

for recursive R, where now "$\Box$" is replaced by "$\exists$" if k is odd and by "$\forall$" if k is even.

The classes Σ_k and Π_k can also be defined by recursion on k. Σ_1 is the class of recursively enumerable relations. Next, a relation belongs to Π_k iff its complement is in Σ_k. And a relation is in Σ_{k+1} iff it is the domain of a relation in Π_k. (We can even start from $k = 0$, by letting Σ_0 be the class of recursive relations.)

> EXAMPLE. The set of Gödel numbers of formulas numeralwise determined by A_E is in Π_2.

> PROOF. a belongs to this set iff [a is the Gödel number of a formula α] and $\forall b \exists d [d$ is $\mathcal{G}$ of a deduction from A_E either of $\alpha(\mathbf{S}^{(b)_0}\mathbf{0}, \mathbf{S}^{(b)_1}\mathbf{0}, \ldots)$ or of the negation of this sentence]. By the technique of Section 3.4, we can show that the phrases in square brackets define recursive relations. By using the English

counterpart to prenex form, we obtain the desired form,

$$\{a \mid \forall b \exists d \, \langle a, b, d \rangle \in R\},$$

with R recursive.   ⊣

One more bit of notation: Let Δ_1 be the class of recursive relations. Then our earlier result (Theorem 35F) stating that a relation is recursive iff both it and its complement are recursively enumerable can now be stated by the equation

$$\Delta_1 = \Sigma_1 \cap \Pi_1.$$

Since this equation holds, we proceed to define Δ_n for $n > 1$ by the analogous equation,

$$\Delta_n = \Sigma_n \cap \Pi_n.$$

The following inclusions hold:

$$
\begin{array}{ccccccc}
 & \Sigma_1 & & \Sigma_2 & & \Sigma_3 & \\
 & \subseteq \quad \supseteq & & \subseteq \quad \supseteq & & \subseteq & \\
\Delta_1 & & \Delta_2 & & \Delta_3 & & \cdots \\
 & \supseteq \quad \subseteq & & \supseteq \quad \subseteq & & \supseteq & \\
 & \Pi_1 & & \Pi_2 & & \Pi_3 &
\end{array}
$$

The case $\Delta_1 \subseteq \Sigma_1$ was mentioned previously (cf. Theorem 35F); its proof hinged on the possibility of "vacuous quantification." The proofs of the other cases are conceptually the same. If x does not occur in φ, then φ, $\forall x \, \varphi$, and $\exists x \, \varphi$ are all equivalent. For example, a relation in Σ_1 is defined by a formula $\exists y \, \varphi$, where φ is numeralwise determined by A_E. But the same relation is defined by $\exists y \, \forall x \, \varphi$ and $\forall x \, \exists y \, \varphi$ (where x does not occur in φ). Hence the relation is also in Σ_2 and Π_2.

It is also true that all the inclusions shown are proper inclusions, i.e., equality does not hold. But we will not prove this fact here. The inclusions are shown pictorially in Fig. 10.

The class of arithmetical relations equals $\bigcup_k \Sigma_k$ and also $\bigcup_k \Pi_k$. For example, any relation in Σ_2 is arithmetical, being defined in $\mathfrak{N}$ by a formula $\exists x \, \forall y \, \varphi$, where φ is numeralwise determined by A_E. Conversely, any arithmetical relation is defined in $\mathfrak{N}$ by some prenex formula. The quantifier-free part of this prenex formula defines a recursive relation (since quantifier-free formulas are numeralwise determined by A_E). Consequently, the defined relation falls somewhere in the hierarchy. The technique of "collapsing" $\exists \exists \cdots \exists$ quantifiers used in the proof of Theorem 35E (and its dual technique for $\forall \forall \cdots \forall$) can be used to good advantage here.
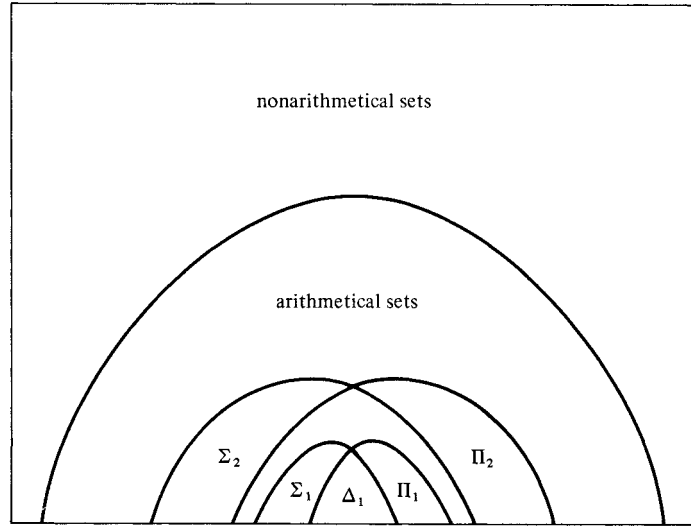
Figure 10. Picture of $\mathcal{PN}$.

Thus we have the following result, which relates definability in $\mathfrak{N}$ to the hierarchy we have just built up from the recursive relations:

THEOREM 35K A relation on the natural numbers is arithmetical (i.e., definable in $\mathfrak{N}$) iff it is in Σ_k for some k, and this property in turn is equivalent to being in Π_l for some l.

In particular, any recursively enumerable relation is arithmetical, as noted previously.

There are certain tricks which are useful in locating specific arithmetical relations in the hierarchy. For example, let A be the set of Gödel numbers of formulas α such that for some n,

$$A_E \vdash \alpha(\mathbf{S}^n\mathbf{0}) \quad \text{and} \quad (\forall i < n)\, A_E \vdash \neg\alpha(\mathbf{S}^i\mathbf{0}).$$

Then $a \in A$ iff [a is the Gödel of a wff α] and $\exists n \exists D[D$ is a deduction from A_E of $\alpha(\mathbf{S}^n\mathbf{0})]$ and $(\forall i < n)(\exists D_i)[D_i$ is a deduction from A_E of $\neg\alpha(\mathbf{S}^i\mathbf{0})]$. The parts in square brackets are recursive so we count the remaining quantifiers. The bounded quantifier "$\forall i < n$" need not be counted. For we have

$$(\forall i < n)(\exists d)\langle d, i\rangle \in P \Leftrightarrow (\exists d)(\forall i < n)\langle (d)_i, i\rangle \in P.$$

Use of this fact lets us push the bounded quantifier inward until it merges with the recursive part. Consequently, $A \in \Sigma_1$.

The following theorem generalizes Theorem 35I.

THEOREM 35L   Let A be a set of sentences such that $\sharp A$ is in Σ_k, where $k > 0$. Then $\sharp \operatorname{Cn} A$ is also in Σ_k.

PROOF.   Return to the proofs of items 18 and 20 of Section 3.4. We had there:

$a \in \sharp \operatorname{Cn} A \Leftrightarrow a$ is the Gödel number of a sentence and $\exists d[d$ is a sequence number and the last component of d is a and for every i less than $\operatorname{lh} d$, either (1) $(d)_i \in \sharp A$, (2) $(d)_i$ is the Gödel number of a logical axiom, or (3) for some j and l less than i, $(d)_j = \langle h(() \rangle * (d)_l * \langle h(\to) \rangle * (d)_i * \langle h() \rangle\rangle]$.

Since $\sharp A \in \Sigma_k$ in (1) we must replace "$(d)_i \in \sharp A$" by something of the form

$$\exists b_1 \forall b_2 \cdots \Box b_k \langle (d)_i, \vec{b}\rangle \in Q$$

for recursive Q. It remains to convert the result into an English prenex expression in Σ_k form. We suggest that the reader set $k = 2$ and write out this expression; the device used in the preceding example will help.   ⊣

Exercises

1. Show that there is no recursive set R such that $\sharp \operatorname{Cn} A_E \subseteq R$ and $\sharp\{\sigma \mid (\neg \sigma) \in \operatorname{Cn} A_E\} \subseteq \overline{R}$, the complement of R. (This result can be stated: The theorems of A_E cannot be recursively separated from the refutable sentences.) *Suggestion*: Make a sentence σ saying "My Gödel number is *not* in R." Look to see where $\sharp\sigma$ is.

2. Let A be a recursive set of sentences in a recursively numbered language with $\mathbf{0}$ and $\mathbf{S}$. Assume that every recursive relation is representable in the theory $\operatorname{Cn} A$. Further assume that A is ω-consistent; i.e., there is no formula φ such that $A \vdash \exists x\, \varphi(x)$ and for all $a \in \mathbb{N}$, $A \vdash \neg\varphi(\mathbf{S}^a\mathbf{0})$. Construct a sentence σ indirectly asserting that it is not a theorem of A, and show that neither $A \vdash \sigma$ nor $A \vdash \neg\sigma$. *Suggestion*: See Section 3.0.

   Remark: This is a version of the incompleteness theorem that is closer to Gödel's original 1931 argument. Note that there is no requirement that the axioms A be *true* in $\mathfrak{N}$. Nor is it required that A include A_E; the fixed-point argument can still be applied.

3. Let T be a theory in a recursively numbered language (with $\mathbf{0}$ and $\mathbf{S}$). Assume that all recursive subsets of $\mathbb{N}$ are weakly representable in T. Show that $\sharp T$ is not recursive. *Suggestion*: Construct a binary relation P such that any weakly representable subset of $\mathbb{N}$ equals $\{b \mid \langle a, b\rangle \in P\}$ for some a, and such that P is recursive if $\sharp T$ is.

Consider the set $H = \{b \mid \langle b, b \rangle \notin P\}$. See Section 3.0. The argument given there for the "diagonal approach" in the special case $T = \text{Th } \mathfrak{N}$ can be adapted here.

Remark: This exercise gives a version of the result, "Any sufficiently strong theory is undecidable."

4. Show that there exist $2^{\aleph_0}$ nonisomorphic countable models of Th $\mathfrak{N}$. *Suggestion*: For each set A of primes, make a model having an element divisible by exactly the primes in A.

5. (Lindenbaum) Let T be a decidable consistent theory (in a reasonable language). Show that T can be extended to a complete decidable consistent theory T'. *Suggestion*: Examine in turn each sentence σ; add either σ or $\neg \sigma$ to T. But take care to maintain decidability.

6. Consider the language of equality, having $\forall$ as its only parameter. Let λ_n be the translation of "There are at least n things," cf. the proof of Theorem 26A. Call a formula *simple* iff it can be built up from atomic formulas and the λ_n's by use of connective symbols (but no quantifiers). Show how, given any formula in the language of equality, we can find a logically equivalent simple formula. *Suggestion*: View this as an elimination-of-quantifiers result (where the quantifiers in λ_n do not count). Use Theorem 31F.

7. (a) Assume that A and B are subsets of $\mathbb{N}$ belonging to Σ_k (or Π_k). Show that $A \cup B$ and $A \cap B$ also belong to Σ_k (or Π_k, respectively).

   (b) Assume that A is in Σ_k (or Π_k) and that the functions $f_1, \ldots, f_m$ are recursive. Show that

$$\{\bar{a} : \langle f_1(\bar{a}), \ldots, f_m(\bar{a}) \rangle \in A\}$$

   is also in Σ_k (or Π_k, respectively). *Suggestion*: First do this for Σ_1. Then observe that the argument used can be generalized.

8. Let T be a theory in a recursively numbered language (with $\mathbf{0}$ and $\mathbf{S}$). Let n be fixed, $n \geq 0$. Assume that all subsets of $\mathbb{N}$ in Σ_n are weakly representable in T. Show that $\sharp T$ is not in Π_n. (Observe that Exercise 3 is a special case of this, wherein $n = 0$. The suggestions given there carry over to the present case.)

9. Show that

$$\{\sharp \sigma \mid A_E; \sigma \text{ is } \omega\text{-consistent}\}$$

(see Exercise 2) is a Π_3 set.

10. The theory Cn A_E has many complete extensions, of which Th $\mathfrak{N}$ is but one. How many? That is, what is the cardinality of the set of complete theories (in the language) that extend A_E?

SECTION 3.6
Recursive Functions

We have used recursive functions (i.e., functions that, when viewed as relations, are recursive) to obtain theorems of incompleteness and undecidability of theories. But the class of recursive functions is also an interesting class in its own right, and in this section we will indicate a few of its properties.

Recall that by Church's thesis, a function is recursive iff it is computable by an effective procedure (page 210). This fact is responsible for much of the interest in recursive functions. At the same time, this fact makes possible an intuitive understanding of recursiveness, which greatly facilitates the study of the subject. Suppose, for example, that you are suddenly asked whether or not the inverse of a recursive permutation of $\mathbb{N}$ is recursive. Before trying to prove this, you should first ask yourself the intuitive counterpart: Is the inverse of a computable permutation f also computable? You then — one hopes — perceive that the answer is affirmative. To compute $f^{-1}(3)$, you can compute $f(0), f(1), \dots$ until for some k it is found that $f(k) = 3$. Then $f^{-1}(3) = k$. Having done this, you have gained two advantages. For one, you feel certain that the answer to the question regarding recursive permutations must also be affirmative. And secondly, you have a good outline of how to prove this; the proof is found by making rigorous the intuitive proof. This strategy for approaching problems involving recursiveness will be very useful in this section.

Before proceeding, it might be wise to summarize here some of the facts about recursive functions we have already established. We know that a function f is recursive iff it (as a relation) is representable in $\text{Cn } A_E$, by Theorem 34A. Consequently, every recursive function is weakly representable in this theory.

In Section 3.3 a repertoire of recursive functions was developed. In addition, it was shown that the class of recursive functions is closed under certain operations, such as composition (Theorem 33L) and the "least-zero" operator (Theorem 33M).

We also know of a few functions that are not recursive. There are uncountably many (to be exact, $2^{\aleph_0}$) functions from $\mathbb{N}^m$ into $\mathbb{N}$ altogether, but only countably many of them can be recursive. So an abundance of nonrecursive functions exists, despite the fact that the most commonly met functions (such as the polynomials) were shown in Section 3.3 to be recursive. By catalog item 1 of Section 3.3, the characteristic function of a nonrecursive set is nonrecursive. For example, if $f(a) = 1$ whenever a is the Gödel number of a member of $\text{Cn } A_E$ and $f(a) = 0$ otherwise, then f is not recursive.

Normal Form

For any computable function, such as the polynomial function $a^2 + 3a + 5$, one can in principle design a digital computer into which one feeds a and out of which comes $a^2 + 3a + 5$ (Fig. 11). But if you then want a different function, you must build a different computer. (Or change the wiring in the one you have.) It was recognized long ago that it is usually more desirable to build a single general-purpose stored-program computer. Into this you feed both a and the program for computing your polynomial (Fig. 12). This "universal" computer requires two inputs, and it will compute any one-place computable function (if supplied with enough memory space), provided that the right program is fed into it. Of course, there are some programs that do not produce any function on $\mathbb{N}$, as many a programmer has, to his sorrow, discovered. (Such programs produce malfunctions instead!)
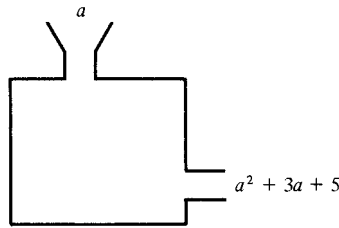
Figure 11. Special-purpose computer.
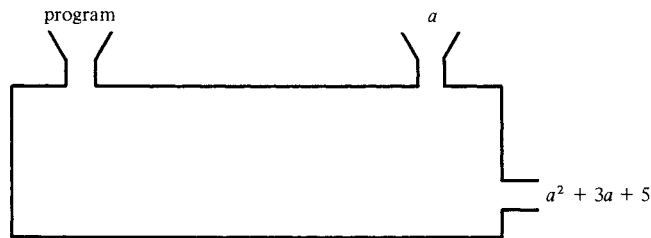
Figure 12. General-purpose computer.

In this subsection and the next, we will repeat what has just been said, but with recursive functions and with proofs. For our universal computer we will have a recursive relation T_1 and a recursive function U. Then for any recursive $f : \mathbb{N} \to \mathbb{N}$ there will exist an e (analogous to the program) such that

$$f(a) = U(\text{the least } k \text{ such that } \langle e, a, k \rangle \in T_1)$$
$$= U(\mu k \, \langle e, a, k \rangle \in T_1),$$

where the second equation is to be understood as being an abbreviation for the first. Actually e will here be the Gödel number of a formula

φ that represents (or at least weakly represents) f in Cn A_E. And the numbers k for which $\langle e, a, k \rangle \in T_1$ will encode both $f(a)$ and $\mathcal{G}$ of a deduction from A_E of $\varphi(\mathbf{S}^a \mathbf{0}, \mathbf{S}^{f(a)} \mathbf{0})$.

DEFINITION.   For each positive integer m, let T_m be the $(m + 2)$-ary relation to which an $(m + 2)$-tuple $\langle e, a_1, \ldots, a_m, k \rangle$ belongs iff

(i) e is the Gödel number of a formula φ in which only $v_1, \ldots, v_m, v_{m+1}$ occur free;

(ii) k is a sequence number of length 2, and $(k)_0$ is $\mathcal{G}$ of a deduction from A_E of $\varphi(\mathbf{S}^{a_1} \mathbf{0}, \ldots, \mathbf{S}^{a_m} \mathbf{0}, \mathbf{S}^{(k)_1} \mathbf{0})$.

The idea here is that for any one-place recursive function f we can first of all take e to be the Gödel number of a formula φ weakly representing f (as a relation). Then we know that for any a and b,

$$A_E \vdash \varphi(\mathbf{S}^a \mathbf{0}, \mathbf{S}^b \mathbf{0}) \quad \text{iff} \quad b = f(a).$$

So any number k meeting clause (ii) of the definition must equal $\langle (k)_0, f(a) \rangle$, where $(k)_0$ is $\mathcal{G}$ of a deduction of $\varphi(\mathbf{S}^a \mathbf{0}, \mathbf{S}^{f(a)} \mathbf{0})$ from A_E. (We have departed from the usual definition of T_m here by not requiring that k be as small as possible.)

Take for the "upshot" function U the function

$$U(k) = (k)_1.$$

This U is recursive and in the situation described in the preceding paragraph we have $U(k) = f(a)$.

LEMMA 36A   For each m, the relation T_m is recursive.

PROOF, FOR $m = 2$. $\langle e, a_1, a_2, k \rangle \in T_2$ iff e is the Gödel number of a formula, $\sharp(\forall v_1 \, \forall v_2 \, \forall v_3) * e$ is the Gödel number of a sentence, k is a sequence number of length 2, and $(k)_0$ is $\mathcal{G}$ of a deduction from A_E of

$$\text{Sb}(\text{Sb}(\text{Sb}(e, \sharp v_1, g(a_1)), \sharp v_2, g(a_2)), \sharp v_3, g((k)_1)),$$

where $g(n) = \sharp \mathbf{S}^n \mathbf{0}$. From Section 3.4 we know all this to be recursive.   ⊣

THEOREM 36B   (a) For any recursive function $f : \mathbb{N}^m \to \mathbb{N}$, there is an e such that for all $a_1, \ldots, a_m$,

$$f(a_1, \ldots, a_m) = U(\mu k \, \langle e, a_1, \ldots, a_m, k \rangle \in T_m).$$

(In particular, such a number k exists.)

(b) Conversely, for any e such that $\forall a_1 \cdots a_m \exists k \, \langle e, a_1, \ldots, a_m, k \rangle \in T_m$, the function whose value at $a_1, \ldots, a_m$ is $U(\mu k \langle e, a_1, \ldots, a_m, k \rangle \in T_m)$ is recursive.

PROOF.   Part (b) follows immediately from the fact that U and T_m are recursive. As for part (a), we take for e the Gödel number of a formula φ weakly representing f in Cn A_E. Given any $\vec{a}$, we know that $A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{f(\vec{a})}\mathbf{0})$. If we let d be $\mathcal{G}$ of a deduction from A_E of this sentence, then $\langle e, \vec{a}, \langle d, f(\vec{a})\rangle\rangle \in T_m$. Hence there is some k for which $\langle e, \vec{a}, k\rangle \in T_m$. And for any such k, we know that $A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{(k)_1}\mathbf{0})$, since $(k)_0$ is $\mathcal{G}$ of a deduction. Consequently, $U(k) = (k)_1 = f(\vec{a})$ by our choice of φ. Thus we have $U(\mu k \, \langle e, \vec{a}, k\rangle \in T_m) = f(\vec{a})$.   ⊣

This theorem, due to Kleene in 1936, shows that every recursive function is representable in the normal form

$$f(\vec{a}) = U(\mu k \, \langle e, \vec{a}, k\rangle \in T_m).$$

Thus a computing machine able to calculate U and the characteristic function of T_1 is a "universal" computer for one-place recursive functions. The input e corresponds to the program, and it must be chosen with care if any output is to result (i.e., if there is to be any k such that $\langle e, a, k\rangle \in T_1$).

Recursive Partial Functions

The theory of recursive functions becomes more natural if we consider the broader context of partial functions.

DEFINITION.   An m-place *partial function* is a function f with dom $f \subseteq \mathbb{N}^m$ and ran $f \subseteq \mathbb{N}$. If $\vec{a} \notin$ dom f, then $f(\vec{a})$ is said to be *undefined*. If dom $f = \mathbb{N}^m$, then f is said to be *total*.

The reader is hereby cautioned against reading too much into our choice of the words "partial" and "total" (or the word "undefined," for that matter). A partial function f may or may not be total; the words "partial" and "total" are not antonyms.

We will begin by looking at those partial functions that are informally computable.

⋆DEFINITION.   An m-place partial function f is *computable* iff there is an effective procedure such that (a) given an m-tuple $\vec{a}$ in dom f, the procedure produces $f(\vec{a})$; and (b) given an m-tuple $\vec{a}$ not in dom f, the procedure produces no output at all.

This definition extends the one previously given for total functions. At that time we proved a result (Theorem 33H), part of which generalizes to partial functions.

⋆**THEOREM 36C**   An m-place partial function f is computable iff f (as an $(m + 1)$-ary relation) is effectively enumerable.

PROOF.   The proof is reminiscent of the proof of another result, Theorem 17E. First suppose we have a way of effectively enumerating f. Given an m-tuple $\vec{a}$, we examine the listing of the relation as the procedure churns it out. If and when an $(m+1)$-tuple beginning with $\vec{a}$ appears, we print out its last component as $f(\vec{a})$.

Conversely, assume that f is computable, and first suppose that f is a one-place partial function. We can enumerate f as a relation by the following procedure:

1. Spend one minute calculating $f(0)$.
2. Spend two minutes calculating $f(0)$, then two minutes calculating $f(1)$.
3. Spend three minutes calculating $f(0)$, three minutes calculating $f(1)$, and three minutes calculating $f(2)$.

And so forth. Of course, whenever one of these calculations produces any output, we place the corresponding pair on the list of members of the relation f.

For a computable m-place partial function, instead of calculating the value of f at $0, 1, 2, \dots$ we calculate its value at $\langle (0)_0, \dots, (0)_{m-1} \rangle$, $\langle (1)_0, \dots, (1)_{m-1} \rangle$, $\langle (2)_0, \dots, (2)_{m-1} \rangle$, etc.   ⊣

In the case of a computable total function f, we were also able to conclude that f was a decidable relation. But this may fail for a nontotal f. For example, let

$$f(a) = \begin{cases} 0 & \text{if } a \in \sharp\mathrm{Cn}\, A_E, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Then f is computable. (We compute $f(a)$ by enumerating $\sharp\mathrm{Cn}\, A_E$ and looking for a.) But f is not a decidable relation, lest $\sharp\mathrm{Cn}\, A_E$ be decidable. On the basis of this example and the foregoing theorem, we select our definition for the precise counterpart of the concept of computable partial function.

DEFINITION.   A *recursive partial function* is a partial function that, as a relation, is recursively enumerable.

The reader should be warned that "recursive partial function" is an indivisible phrase; a recursive partial function need *not* be (as relation) recursive. But at least for a total function our terminology is consistent with past practice.

THEOREM 36D   Let $f : \mathbb{N}^m \to \mathbb{N}$ be a total function. Then f is a recursive partial function iff f is recursive (as a relation).

PROOF.   If f is recursive (as a relation), then *a fortiori* f is recursively enumerable. Conversely, suppose that f is recursively

enumerable. Since f is total,

$$f(\vec{a}) \neq b \iff \exists c[f(\vec{a}) = c \ \& \ b \neq c].$$

The form of the right-hand side shows that the complement of f is also recursively enumerable. Thus by Theorem 35F, f is recursive.   ⊣

In first discussing normal form results, we pictured a two-input device (Fig. 13). For any computable partial function, there is some program that computes it. But now the converse holds: Any program will produce some computable *partial* function. Of course many programs will produce the empty function, but that is a computable partial function.
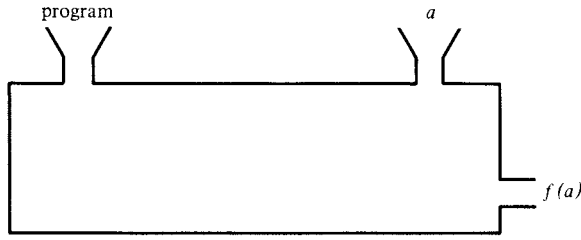
Figure 13. Computer with program for f.

For the recursive partial functions the same considerations apply. Define, for each $e \in \mathbb{N}$, the m-place partial function $[\![e]\!]_m$ by

$$[\![e]\!]_m(a_1, \ldots, a_m) = U(\mu k \, \langle e, a_1, \ldots, a_m, k \rangle \in T_m).$$

The right-hand side is to be understood as undefined if there is no such k. In other words,

$$\vec{a} \in \mathrm{dom}[\![e]\!]_m \quad \text{iff} \quad \exists k \, \langle e, a_1, \ldots, a_m, k \rangle \in T_m,$$

in which case the value $[\![e]\!]_m(\vec{a})$ is given by the above equation.

The following theorem is an improved version of Theorem 36B:

NORMAL FORM THEOREM (KLEENE, 1943) (a) The $(m+1)$-place partial function whose value at $\langle e, a_1, \ldots, a_m \rangle$ is $[\![e]\!]_m(a_1, \ldots, a_m)$ is a recursive partial function.

(b) For each $e \geq 0$, $[\![e]\!]_m$ is an m-place recursive partial function.

(c) Any m-place recursive partial function equals $[\![e]\!]_m$ for some e.

PROOF. (a) We have

$$[\![e]\!]_m(\vec{a}) = b \iff \exists k[\langle e, \vec{a}, k \rangle \in T_m \ \& \ U(k) = b \ \& \ (\forall k' < k)\langle e, \vec{a}, k' \rangle \notin T_m].$$

The part in square brackets is recursive, so the function (as a relation) is recursively enumerable.

(b) The above proof still applies, e now being held fixed.

(c) Let f be an m-place recursive partial function, so that $\{\langle \vec{a}, b \rangle \mid f(\vec{a}) = b\}$ is recursively enumerable. Hence there is a formula φ that weakly represents this relation in Cn A_E. We claim that $f = [\![\sharp \varphi]\!]_m$. For if $f(\vec{a}) = b$, then $A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \dots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^b\mathbf{0})$. Hence there is a k such that $\langle \sharp \varphi, \vec{a}, k \rangle \in T_m$. For any such k, $U(k) = b$, since $A_E \nvdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \dots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^c\mathbf{0})$ for $c \neq b$. Similarly, if $f(\vec{a})$ is undefined, then $A_E \nvdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \dots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^c\mathbf{0})$ for any c, whence $[\![\sharp \varphi]\!]_m$ is undefined here also. $\dashv$

Part (a) of the normal form theorem (in the case $m = 1$) tells us that the function Φ defined by the equation

$$\Phi(e, a) = [\![e]\!]_1(a) = U(\mu k \, \langle e, a, k \rangle \in T_1)$$

is a recursive partial function. And part (c) tells us that Φ is "universal" in the sense that we can get any one-place recursive partial function from Φ by holding the first variable fixed at a suitable value.

The informal counterpart of the universal function Φ is the computer operating system. The operating system takes two inputs, the program e and the data a. And it runs the program on that data. But the operating system itself is *computable* as a two-place partial function.

The proof of normal form theorem gives us a way to compute the values of our "operating system" Φ, albeit in an extremely inefficient way. The straightforward idea of "looking at the program e and doing what it says to the data a" has been obscured, to say the least.

The function $[\![e]\!]_m$ is said to be the m-place recursive partial function with *index* e. Part (c) of the normal form theorem tells us that every recursive partial function has an index. The proof shows that the Gödel number of a formula weakly representing a function is always an index of the function.

We now have a convenient indexing $[\![0]\!]_1, [\![1]\!]_1, \dots$ of all the one-place recursive partial functions. Function $[\![e]\!]_1$ is produced by the "instructions" encoded by e. Of course, that function will be empty unless e is the Gödel number of a formula and certain other conditions are met.

All the recursive total functions are included in our enumeration of recursive partial functions. But we cannot tell effectively by looking at a number e whether or not it is the index of a total function:

THEOREM 36E $\{e \mid [\![e]\!]_1 \text{ is total}\}$ is not recursive.

PROOF.   Call this set A. Consider the function defined by

$$f(a) = \begin{cases} [\![a]\!]_1(a) + 1 & \text{if } a \in A, \\ 0 & \text{if } a \notin A. \end{cases}$$

Then f, by its construction, is total. Is it recursive? We have

$$f(a) = b \iff [(a \notin A \ \& \ b = 0) \ \text{or} \ (a \in A \ \& \ \exists k (\langle a, a, k \rangle \in T_1 \\ \& \ b = U(k) + 1 \ \& \ (\forall j < k)\langle a, a, j \rangle \notin T_1))].$$

Thus if A is recursive, then f (as a relation) is recursively enumerable. But then f is a total recursive function, and so equals $[\![e]\!]_1$ for some $e \in A$. But $f(e) = [\![e]\!]_1(e) + 1$, so we cannot have $f = [\![e]\!]_1$. This contradiction shows that A cannot be recursive.

⊣

It is not hard to show that A is in Π_2. This classification is the best possible, as it can be shown that A is not in Σ_2.

THEOREM 36F   The set

$$K = \{a \mid [\![a]\!]_1(a) \text{ is defined}\}$$

is recursively enumerable but not recursive.

PROOF. K is recursively enumerable, since $a \in K \Leftrightarrow \exists k \, \langle a, a, k \rangle \in T_1$. To see that K cannot be recursive, consider the function defined by

$$g(a) = \begin{cases} [\![a]\!]_1(a) + 1 & \text{if } a \in K, \\ 0 & \text{if } a \notin K. \end{cases}$$

This is a total function. Exactly as in the preceding theorem, we have that K cannot be recursive.   ⊣

COROLLARY 36G (UNSOLVABILITY OF THE HALTING PROBLEM)   The relation

$$\{\langle e, a \rangle \mid [\![e]\!]_1(a) \text{ is defined}\}$$

is not recursive.

PROOF.   We have $a \in K$ iff the pair $\langle a, a \rangle$ belongs to this relation. (Thus the problem of membership in K is "reducible" to the halting problem.) If this relation were recursive, then K would be, which is not the case.   ⊣

This corollary tells us that there is no effective way to tell, given a program e for a recursive partial function and an input a, whether or not the function $[\![e]\!]_1$ is defined at a.

We can obtain an indexing of the recursively enumerable relations by using the following characterization.

THEOREM 36H   A relation on $\mathbb{N}$ is recursively enumerable iff it is the domain of some recursive partial function.

PROOF.   The domain of any recursively enumerable relation is also recursively enumerable; cf. part 4 of Theorem 35E. In particular,

the domain of any recursive partial function is recursively enumerable.

Conversely, let Q be any recursively enumerable relation, where

$$\vec{a} = Q \Leftrightarrow \exists b \, \langle \vec{a}, b \rangle \in R$$

with R recursive. Let

$$f(\vec{a}) = \mu b \, \langle \vec{a}, b \rangle \in R;$$

i.e.,

$$f(\vec{a}) = b \iff \langle \vec{a}, b \rangle \in R \ \& \ (\forall c < b) \langle \vec{a}, c \rangle \notin R.$$

Then f, as a relation, is recursive. Hence f is a recursive partial function. Clearly its domain is Q.   ⊣

Thus our indexing of the recursive partial functions induces an indexing of the recursively enumerable relations. Define

$$W_e = \text{dom}[\![e]\!]_1.$$

Then $W_0, W_1, W_2, \ldots$ is a list of all recursively enumerable subsets of $\mathbb{N}$. In Theorem 36E we showed that $\{e \mid W_e = \mathbb{N}\}$ is not recursive. Similarly, Theorem 36F asserts that $\{e \mid e \in W_e\}$ is not recursive. Define a relation Q by

$$Q = \{\langle e, a \rangle \mid a \in W_e\}.$$

Then Q is recursively enumerable, since $\langle e, a \rangle \in Q \Leftrightarrow \exists k \, \langle e, a, k \rangle \in T_1$. Furthermore, Q is universal for recursively enumerable sets, in the sense that for any recursively enumerable $A \subseteq \mathbb{N}$ there is some e such that $A = \{a \mid \langle e, a \rangle \in Q\}$. The unsolvability of the halting problem can be stated: Q is not recursive.

We can apply the classical diagonal argument to "diagonalize out" of the list $W_0, W_1, W_2, \ldots$ of recursively enumerable sets. The set $\{a \mid a \notin W_a\}$ cannot coincide with any W_q. In fact this set is exactly $\overline{K}$, the complement of the set K in Theorem 36F. Because

$$q \in \overline{K} \iff q \notin W_q,$$

the set $\overline{K}$ cannot equal any W_q; the number q witnesses the inequality of the two sets $\overline{K}$ and W_q.

And there is more: Whenever W_q is a recursively enumerable *subset* of $\overline{K}$, that is, $W_q \subseteq \overline{K}$, then we can produce a number in $\overline{K}$ that is not in W_q. Such a number is q itself. To see this, observe that in the line displayed in the preceding paragraph we cannot have both sides false ($q \in K$ and $q \in W_q$) because $W_q \subseteq \overline{K}$. So both sides are true.

Theorem 36F asserts that K, although recursively enumerable, is not recursive. To show non-recursiveness, it suffices to show that its complement $\overline{K}$ is not recursively enumerable. The preceding paragraph

does this in a particularly strong way, thereby giving us a second proof of Theorem 36F.

At this point, let us reconsider the Gödel incompleteness theorem, from the computability point of view.

The set K is recursively enumerable (i.e., Σ_1). It follows (cf. Theorem 35K) that K is arithmetical; that is, K is definable in the structure $\mathfrak{N}$.

So there is a formula $\varkappa(v_1)$ with just v_1 free that defines K in $\mathfrak{N}$. And so the set $\overline{K}$ is defined in $\mathfrak{N}$ by the formula $\neg\varkappa(v_1)$. Thus we have

$$a \in \overline{K} \iff (\neg\varkappa(\mathbf{S}^a\mathbf{0})) \in \mathrm{Th}\,\mathfrak{N}.$$

This fact tells how we can "reduce" questions about membership in the set $\overline{K}$ to questions about $\mathrm{Th}\,\mathfrak{N}$. Imagine that we are given a number a, and we want to know whether or not $a \in \overline{K}$. We can *compute* the number $\sharp(\neg\varkappa(\mathbf{S}^a\mathbf{0}))$. (Informally, it is clear that we can *effectively* compute this number. Formally, we apply item 5 from Section 3.4 to make sure we can *recursively* compute the number.) If we somehow had an oracle for $\sharp\,\mathrm{Th}\,\mathfrak{N}$ (i.e., a magic device that, given a number, would tell us whether or not that number was in $\sharp\,\mathrm{Th}\,\mathfrak{N}$), then we could answer the question "Is $a \in \overline{K}$?"

Now let us eliminate the magic. For sets A and B of natural numbers, we say that A is *many-one reducible* to B (in symbols, $A \leq_{\mathrm{m}} B$) iff there exists a total recursive function f such that for every number a,

$$a \in A \iff f(a) \in B.$$

The earlier example tells us that $\overline{K} \leq_{\mathrm{m}} \sharp\,\mathrm{Th}\,\mathfrak{N}$. More generally, the argument shows that any arithmetical set is many-one reducible to $\sharp\,\mathrm{Th}\,\mathfrak{N}$.

> **LEMMA 36I**   Assume that A and B are sets of natural numbers with $A \leq_{\mathrm{m}} B$.
>
>   (a) If B is recursive, then A is also recursive.
>   (b) If B is recursively enumerable, the A is also recursively enumerable.
>   (c) If B is Σ_n for some n, the A is also Σ_n for that n.

> PROOF.   Part (a) is already familiar; it was, in different terminology, catalog item 2 in Section 3.3.
>
>   Part (b) is essentially part (a) "plus a quantifier." That is, because B is recursively enumerable, we know that for some recursive binary relation Q,
>
> $$c \in B \iff \exists b\, Q(c, b).$$
>
>   If f is the total recursive function that many-one reduces A to B, then every number a,
>
> $$a \in A \iff f(a) \in B \iff \exists b[Q(f(a), b)].$$

The part in square brackets is recursive (i.e., $\{\langle a, b \rangle \mid Q(f(a), b)\}$ is recursive), as in part (a) of the lemma. So we have A in the required form to be recursively enumerable.

Part (c) is essentially part (a) "plus n quantifiers" and is proved like part (b).   ⊣

Our reason for examining the particular set $\overline{K}$ is that it gives us the following consequence:

GÖDEL INCOMPLETENESS THEOREM   Th $\mathfrak{N}$ is not recursively axiomatizable.

PROOF.   Th $\mathfrak{N}$ cannot be recursively enumerable, lest $\overline{K}$ be recursively enumerable, by the preceding lemma. But any recursively axiomatizable theory would be recursively enumerable (item 20 of Section 3.4; also Theorem 35I).   ⊣

In starkest terms, the situation is this: Any recursively axiomatizable theory is recursively enumerable. But Th $\mathfrak{N}$ is not recursively enumerable. So any recursively axiomatizable subtheory must be incomplete.

It will be worth while to go over this proof again, but replacing negative statements (such-and-such a set does *not* have a particular property) by positive statements.

Assume that T is any recursively axiomatizable subtheory of Th $\mathfrak{N}$. (So by the above theorem, T is incomplete.) We want to lay our hands on a sentence demonstrating the incompleteness.

We have made a total recursive function f that many-one reduces $\overline{K}$ to $\sharp$ Th $\mathfrak{N}$, namely $f(a) = \sharp(\neg \varkappa(\mathbf{S}^a\mathbf{0}))$; then for every a,

$$a \in \overline{K} \iff f(a) \in \sharp \,\text{Th}\, \mathfrak{N}.$$

And $f(a)$ is (the Gödel number of) the sentence saying "$a \notin K$."

Consider the set J of numbers defined by the condition

$$a \in J \iff f(a) \in \sharp T.$$

Thus J is the set of numbers that T "knows" are not in K. There are two observations to be made concerning J:

First, J is recursively enumerable. It is many-one reduced by f to the recursively enumerable set $\sharp T$; apply Lemma 36I(b).

Secondly, $J \subseteq \overline{K}$. We have $T \subseteq \text{Th}\, \mathfrak{N}$, so if T knows that $a \notin K$, then really $a \notin K$:

$$a \in J \iff f(a) \in \sharp T \implies f(a) \in \sharp \,\text{Th}\, \mathfrak{N} \iff a \in \overline{K}.$$

So J is a recursively enumerable subset of $\overline{K}$. It is a *proper* subset, because $\overline{K}$ is not recursively enumerable. That is, there is some number q with $q \in \overline{K}$ and $q \notin J$. Consequently, $f(q) \in \sharp \,\text{Th}\, \mathfrak{N}$ but $f(q) \notin \sharp T$. That is, the sentence $(\neg \varkappa(\mathbf{S}^q\mathbf{0}))$ is true (in $\mathfrak{N}$) but fails to be in T, thereby demonstrating the incompleteness of T.

And what does this sentence "say"? For q, we can take any number for which $W_q = J$. Then $q \in \overline{K}$ and $q \notin J$.

Here then is the situation:

$$(\neg \varkappa(\mathbf{S}^q \mathbf{0})) \quad \text{says } q \notin K$$
$$\text{i.e., } q \notin W_q$$
$$\text{i.e., } q \notin J \qquad \text{since } W_q = J$$
$$\text{i.e., } f(q) \notin \sharp T \qquad \text{by definition of } J$$
$$\text{i.e., } T \nvdash (\neg \varkappa(\mathbf{S}^q \mathbf{0})).$$

The sentence we made to witness the incompleteness of T asserts its own unprovability in the axiomatizable theory T!

The computability approach and the self-reference approach to Gödel's incompleteness theorem are not so different after all. Moreover, the computability approach is close to the diagonalization approach (of Section 3.0), but with the diagonal argument moved to a different context.

Reduction of Decision Problems[1]

Suppose we have a two-place recursive partial function f. Then we claim that, for example, the function g defined by

$$g(a) = f(3, a)$$

is also a recursive partial function. On the basis of informal computability this is clear; one computes g by plugging in 3 for the first variable and then following the instructions for f. A proof can be found by formalizing this argument. There is some formula $\varphi = \varphi(v_1, v_2, v_3)$ that weakly represents f (as a relation) in Cn A_E. Then g is weakly represented by $\varphi(\mathbf{S}^3 \mathbf{0}, v_1, v_2)$, provided that v_1 and v_2 are substitutable in φ for v_2 and v_3. (If not, we can always use an alphabetic variant of φ.)

Now all this is not very deep. But by standing back and looking at what was said, we perceive a more subtle fact. We were able to transform effectively the instructions for f into instructions for g. So there should be a recursive function that, given an index for f and the number 3, will produce an index for g. The following formulation of this fact is sometimes known by the cryptic name of "the S-m-n theorem."

> **PARAMETER THEOREM**   For each $m \geq 1$ and $n \geq 1$ there is a recursive function ρ such that for any $e, \vec{a}, \vec{b}$,

$$[\![e]\!]_{m+n}(a_1, \ldots, a_m, b_1, \ldots, b_n) = [\![\rho(e, a_1, \ldots, a_m)]\!]_n(b_1, \ldots, b_n).$$

(Equality here means of course that if one side is defined, then so also is the other side, and the values coincide. Sometimes a special symbol "$\simeq$" is used for this role.)

---

[1] The remainder of this section can be skipped on a first reading.

On the left side of the equation $\vec{a}$ consists of arguments for the function $[\![e]\!]_{m+n}$; on the right side $\vec{a}$ consists of parameters upon which the function $[\![\rho(e, \vec{a})]\!]_n$ depends. In the example we had $m = n = 1$ and $a_1 = 3$. Since ρ depends on m and n, the notation "ρ_n^m" would be logically preferable. But, in fact, we will use simply "ρ."

PROOF, FOR $m = n = 1$.   It is possible to give a proof along the lines indicated by the discussion that preceded the theorem. But to avoid having to cope with alphabetic variants, we will adopt a slightly different strategy.

We know from the normal form theorem that the three-place partial function h defined by

$$h(e, a, b) = [\![e]\!]_2(a, b)$$

is a recursive partial function. Hence there is a formula ψ that weakly represents h (as a relation). We may suppose that in ψ the variables v_1 and v_2 are not quantified. We can then take

$$\rho(e, a) = \sharp\psi\,(\mathbf{S}^e\mathbf{0}, \mathbf{S}^a\mathbf{0}, v_1, v_2)$$
$$= \mathrm{Sb}(\mathrm{Sb}(\mathrm{Sb}(\mathrm{Sb}(\sharp\psi, \sharp v_1, \sharp\mathbf{S}^e\mathbf{0}), \sharp v_2, \sharp\mathbf{S}^a\mathbf{0}), \sharp v_3, \sharp v_1), \sharp v_4, \sharp v_2).$$

Then $\rho(e, a)$ is the Gödel number of a formula weakly representing the function $g(b) = [\![e]\!]_2(a, b)$. Hence it is an index of g.
$$\dashv$$

We will utilize the parameter theorem to show that certain sets are *not* recursive. We already know that $K = \{a \mid [\![a]\!]_1(a)$ is defined$\}$ is not recursive. For a given nonrecursive set A we can sometimes find a (total) recursive function g such that

$$a \in K \Leftrightarrow g(a) \in A$$

or a (total) recursive function g' such that

$$a \notin K \Leftrightarrow g'(a) \in A.$$

In either case it then follows at once that A cannot be recursive lest K be. In the former case we have $K \leq_m A$ and A is not Π_1 (by Lemma 36I); in the latter case $\overline{K} \leq_m A$ and A is not Σ_1. In either case, A is not recursive. The function g or g' can often be obtained from the parameter theorem.

EXAMPLE. $\{a \mid W_a = \varnothing\}$ is not recursive.

PROOF.   Call this set A. First, note that $A \in \Pi_1$, since $W_a = \varnothing$ iff $\forall b \forall k \langle a, b, k \rangle \notin T_1$. Consequently, K cannot be many-one reducible to A, but it is reasonable to hope that $\overline{K}$ might be. That is, we want a total recursive function g such that

$$[\![a]\!]_1(a) \text{ is undefined} \quad \Leftrightarrow \quad \mathrm{dom}[\![g(a)]\!]_1 = \varnothing.$$

This will hold if for all b, $[\![g(a)]\!]_1(b) = [\![a]\!]_1(a)$. So start with the recursive partial function

$$f(a, b) = [\![a]\!]_1(a)$$

and let $g(a) = \rho(\hat{f}, a)$, where $\hat{f}$ is an index for f. Then

$$[\![g(a)]\!]_1(b) = [\![\rho(\hat{f}, a)]\!]_1(b) = f(a, b) = [\![a]\!]_1(a).$$

Thus this g shows that $\overline{K}$ is many-one reducible to A.   ⊣

THEOREM 36J (RICE, 1953)   Let $\mathcal{C}$ be a set of one-place recursive partial functions. Then the set $\{e \mid [\![e]\!]_1 \in \mathcal{C}\}$ of indices of members of $\mathcal{C}$ is recursive iff either $\mathcal{C}$ is empty or $\mathcal{C}$ contains all one-place recursive partial functions.

PROOF.   Only one direction requires proof. Let $I_\mathcal{C} = \{e \mid [\![e]\!]_1 \in \mathcal{C}\}$ be the set of indices of members of $\mathcal{C}$.

Case I: The empty function $\varnothing$ is not in $\mathcal{C}$. If nothing at all is in $\mathcal{C}$ we are done, but suppose some function ψ is in $\mathcal{C}$. We can show that K is many-one reducible to $I_\mathcal{C}$ if we have a recursive total function g such that

$$[\![g(a)]\!]_1 = \begin{cases} \psi & \text{if } a \in K, \\ \varnothing & \text{if } a \notin K. \end{cases}$$

For then $a \in K \Leftrightarrow [\![g(a)]\!]_1 \in \mathcal{C} \Leftrightarrow g(a) \in I_\mathcal{C}$.
We can obtain g from the parameter theorem by defining

$$g(a) = \rho(e, a),$$

where

$$[\![e]\!]_2(a, b) = \begin{cases} \psi(b) & \text{if } a \in K, \\ \text{undefined} & \text{if } a \notin K. \end{cases}$$

The above *is* a recursive partial function, since

$$[\![e]\!]_2(a, b) = c \Leftrightarrow a \in K \ \& \ \psi(b) = c$$

and the right-hand side is recursively enumerable.
Case II: $\varnothing \in \mathcal{C}$. Then apply case I to the complement $\overline{\mathcal{C}}$ of $\mathcal{C}$. We can then conclude that $I_{\overline{\mathcal{C}}}$ is not recursive. But $I_{\overline{\mathcal{C}}}$ is the complement of $I_\mathcal{C}$, so $I_\mathcal{C}$ cannot be recursive.
Thus in either case, $I_\mathcal{C}$ is not recursive.   ⊣

EXAMPLES.   For any fixed e, the set $\{a \mid W_a = W_e\}$ is not recursive, as a consequence of Rice's theorem. In particular, $\{a \mid W_a = \varnothing\}$ is not recursive, a result proved in an earlier example. For two other applications of Rice's theorem, we can say that $\{a \mid W_a \text{ is infinite}\}$ and $\{a \mid W_a \text{ is recursive}\}$ are not recursive.

Register Machines

There are many equivalent definitions of the class of recursive functions. Several of these definitions employ idealized computing devices. These computing devices are like digital computers but are free of any limitation on memory space. The first definition of this type was published by Alan Turing in 1936; similar work was done by Emil Post at roughly the same time. We will give here a variation on this theme, due to Shepherdson and Sturgis (1963).

A *register machine* will have a finite number of registers, numbered $1, 2, \ldots, K$. Each register is capable of storing a natural number of any magnitude. The operation of the machine will be determined by a *program*. A program is a finite sequence of *instructions*, drawn from the following list:

I r (where $1 \leq r \leq K$). "Increment r." The effect of this instruction is to increase the contents of register r by 1. The machine then proceeds to the next instruction in the program.

D r (where $1 \leq r \leq K$). "Decrement r." The effect of this instruction depends on the contents of register r. If that number is nonzero, it is decreased by 1 and the machine then proceeds, not to the next instruction, but to the following one. But if the number in register r is zero, the machine just proceeds to the next instruction. In summary: The machine tries to decrement register r and skips an instruction if it is successful.

T q (where q is an integer–positive, negative, or zero). "Transfer q." All registers are left unchanged. The machine takes as its next instruction the qth instruction following this one in the program (if $q \geq 0$), or the $|q|$th instruction preceding this one (if $q < 0$). The machine halts if there is no such instruction in the program. An instruction of T 0 results in a loop, with the machine executing this one instruction over and over again.

EXAMPLES

1. Program to clear register 7.

>D 7   Try to decrement 7.
 T 2
 T −2   Go back and repeat.
                  Halt.

2. Program to move a number from register r to register s.

Clear register s (Use the program of the first example.)
>D r Take 1 from r.
 T 3   Halt when zero.
 I s Add 1 to s.
 T −3   Repeat.

This program has seven instructions altogether. It leaves a zero in register r.

3. Program to add register 1 to registers 2 and 3.

$$
\begin{array}{l}
\rightarrow\text{D} \quad 1 \\
\quad\text{T} \quad 4 \\
\quad\text{I} \quad 2 \\
\quad\text{I} \quad 3 \\
\quad\text{T} \ -4
\end{array}
$$
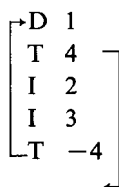
4. (Addition) Say that a and b are in registers 1 and 2. We want $a + b$ in register 3, and we want to leave a and b still in registers 1 and 2 at the end.

	Register contents			
Clear register 3.	a	b	0	
Move number from register 1 to register 4.	0	b	0	a
Add register 4 to registers 1 and 3.	a	b	a	0
Move number from register 2 to register 4.	a	0	a	b
Add register 4 to registers 2 and 3.	a	b	$a+b$	0

This program has 27 instructions as it is written, but three of them are unnecessary. (In the fourth line we begin by clearing register 4, which is already clear.) At the end we have the number a back in register 1. But during the program register 1 must be cleared; this is the only way of determining the number a.

5. (Subtraction) Let $a \,\dot{-}\, b = \max(a - b, 0)$. We leave this program to the reader (Exercise 11).

Now suppose f is an n-place partial function on $\mathbb{N}$. Possibly there will be a program P such that if we start a register machine (having all the registers to which P refers) with $a_1, \ldots, a_n$ in registers $1, \ldots, n$ and apply program P, then the following conditions hold:

(i) If $f(a_1, \ldots, a_n)$ is defined, then the calculation eventually terminates with $f(a_1, \ldots, a_n)$ in register $n + 1$. Furthermore, the calculation terminates by seeking a $(p+1)$st instruction, where p is the length of P.

(ii) If $f(a_1, \ldots, a_n)$ is undefined, then the calculation never terminates.

If there is such a program P, we say that P *calculates* f.

THEOREM 36K Let f be a partial function. Then there is a program that calculates f iff f is a recursive partial function.

Thus by using register machines we arrive at exactly the class of recursive partial functions, a class we originally defined in terms of representability in consistent finitely axiomatizable theories. The fact

that such different approaches produce the same class of partial functions is evidence that this is a significant class.

OUTLINE OF PROOF.   To show that the functions calculable by register machines are recursive partial functions, one "arithmetizes calculations" in the same spirit as we arithmetized deductions in Section 3.4. That is, one assigns Gödel numbers to programs and to sequences of memory configurations. One then verifies that the relevant concepts, translated into numerical relations by the Gödel numbering, are all recursive. (After going through this, one perceives that, from a sufficiently general point of view, deductions and calculations are really about the same sort of thing.)

Conversely, to show that the recursive partial functions are calculable by register machines, one can work through Sections 3.3 and 3.4 again, but where functions were previously shown to be representable in Cn A_E, they must now be shown to be calculable by register machines. This is not as hard as it might sound, since after the first few pages, the proofs are all the same as the ones used before. There is a reason for this similarity. It can be shown that the class of all recursive functions is generated from a certain handful of recursive functions by the operation of composition (in the sense of Theorem 33L) and the "least-zero" operator (Theorem 33M). Much of the work in Sections 3.3 and 3.4 amounts to a verification of this fact. Thus once one has shown that each function in this initial handful is calculable by a register machine and that the class of functions calculable by register machines is closed under composition and the least-zero operator, then the earlier work can be carried over, yielding the calculability of all recursive functions.   ⊣

Exercises

1. Define functions f and g by

$$f(n) = \begin{cases} 0 & \text{if Goldbach's conjecture is true,} \\ 1 & \text{otherwise;} \end{cases}$$

$$g(n) = \begin{cases} 0 & \text{if in the decimal expansion of } \pi \text{ there} \\ & \text{is a run of at least } n \text{ consecutive 7's,} \\ 1 & \text{otherwise.} \end{cases}$$

Is f recursive? Is g recursive? (Goldbach's conjecture says that every even integer greater than 2 is the sum of two primes. The first edition of this book used Fermat's last theorem here.)

2. Define the "diagonal" function $d(a) = [\![a]\!]_1(a) + 1$.
   (a) Show that d is a recursive partial function.
   (b) By part (a), we have $d = [\![e]\!]_1$ for a certain number e. So on the one hand, $d(e) = [\![e]\!]_1(e)$ and on the other hand $d(e) = [\![e]\!]_1(e) + 1$. Can we cancel, to conclude that $0 = 1$? *Suggestion*: Use the special symbol "$\simeq$" to mean that either both sides of the equation are undefined, or both sides are defined and equal. Restate the argument in this notation.

3. (a) Show that the range of any recursive partial function is recursively enumerable.
   (b) Show that the range of a strictly increasing (i.e., $f(n) < f(n+1)$) total recursive function f is recursive.
   (c) Show that the range of a nondecreasing (i.e., $f(n) \le f(n+1)$) total recursive function f is recursive.

4. (a) Let A be a nonempty recursively enumerable subset of $\mathbb{N}$. Show that A is the range of some total recursive function.
   (b) Show that any infinite recursively enumerable subset of $\mathbb{N}$ includes an infinite recursive subset.

5. Show that every recursive partial function has infinitely many indices.

6. Give an example of a function f and a number e such that for all a,

$$f(a) = U(\mu k \langle e, a, k \rangle \in T_1)$$

but e is not the Gödel number of a formula weakly representing in $\operatorname{Cn} A_E$.

7. Show that the parameter theorem can be strengthened by requiring ρ to be one-to-one.

8. Recall that the union of two recursively enumerable sets is recursively enumerable (Exercise 7 of Section 3.5). Show that there is a total recursive function g such that $W_{g(a,b)} = W_a \cup W_b$.

9. Show that $\{a \mid W_a$ has two or more members$\}$ is in Σ_1 but not in Π_1.

10. Show that there is no recursively enumerable set A such that $\{[\![a]\!]_1 \mid a \in A\}$ equals the class of total recursive functions on $\mathbb{N}$.

11. Give register machine programs that calculate the following functions:
    (a) Subtraction, $a \dot{-} b = \max(a - b, 0)$.
    (b) Multiplication, $a \cdot b$.
    (c) $\max(a, b)$.

12. Assume that there is a register machine program that calculates the n-place partial function f. Show that given any positive

integers $r_1, \ldots, r_n$ (all distinct), p, and k, we can find a program Q such that whenever we start a register machine (having all the registers to which Q refers) with $a_1, \ldots, a_n$ in registers $r_1, \ldots, r_n$ and apply program Q, then (i) if $f(a_1, \ldots, a_n)$ is defined, then the calculation eventually terminates with $f(a_1, \ldots, a_n)$ in register p, with the contents of registers $1, 2, \ldots, k$ (except for register p) the same as their initial contents, and furthermore the calculation terminates by seeking a $(q+1)$st instruction, where q is the length of Q; (ii) if $f(a_1, \ldots, a_n)$ is undefined, then the calculation never terminates.

13. Let $g : \mathbb{N}^{n+1} \to \mathbb{N}$ be a (total) function that is calculated by some register machine program. Let $f(a_1, \ldots, a_n) = \mu b[g(a_1, \ldots, a_n, b) = 0]$, where the right-hand side is undefined if no such b exists. Show that the partial function f can be calculated by some register machine program.

14. Show that the following sets have the given location in the arithmetical hierarchy. (In each case, the given location is the best possible, but we will not prove that fact.)
    (a) $\{e \mid [\![e]\!]_1 \text{ is total}\}$ is Π_2.
    (b) $\{e \mid W_e \text{ is finite}\}$ is Σ_2.
    (c) $\{e \mid W_e \text{ is cofinite}\}$ is Σ_3.
    (d) $\{e \mid W_e \text{ is recursive}\}$ is Σ_3.

15. Let $Tot = \{e \mid [\![e]\!]_1 \text{ is total}\}$. Clearly $Tot \subset K$. Show that there is no recursive set A with

$$Tot \subseteq A \subseteq K.$$

Remark: This result includes Theorems 36E and 36F; the proofs used there can be adapted here.

16. (a) Show that each Π_2 set of natural numbers is, for some number e, the set

$$\{a \mid \forall b \exists c \, T_2(e, a, b, c)\}.$$

   (b) Show that the set $\{a \mid \text{not } \forall b \exists c \, T_2(a, a, b, c)\}$ is Σ_2 but not Π_2.
   *(c) Generalize parts (a) and (b) to show that for each n, there is a set that is Σ_n but not Π_n.

17. Assume that A is a set of natural numbers that is arithmetical but is not Π_m. Use the argument of page 256 to show that $\sharp \, \mathrm{Th}\, \mathfrak{N}$ is not Σ_m.

   Remark: Exercises 16 and 17 give a proof of Tarski's theorem (that $\sharp \, \mathrm{Th}\, \mathfrak{N}$ is not arithmetical) from computability theory.

SECTION 3.7
Second Incompleteness Theorem

Let us return once again to item 20 in Section 3.4. Suppose that we have a recursively axiomatizable theory T given by a recursive set A of axioms (i.e., $\sharp A$ is recursive). Then as in item 20

$$a \in \sharp T \iff \exists d \, [d \text{ is the number of a deduction from } A$$
$$\text{and the last component of } d \text{ is } a \text{ and } a$$
$$\text{is the Gödel number of a sentence}].$$

The set of pairs $\langle a, d \rangle$ meeting the condition in brackets is recursive; let $\pi(v_1, v_2)$ be a formula — chosen in some natural way — that numeralwise represents that binary relation in A_E.

For any sentence σ, we can express "$T \vdash \sigma$" by the sentence $\exists \, v_2 \, \pi(\mathbf{S}^{\sharp\sigma}\mathbf{0}, v_2)$. Let us give that sentence a name; define

$$\mathrm{Prb}_T \, \sigma = \exists \, v_2 \pi(\mathbf{S}^{\sharp\sigma}\mathbf{0}, v_2).$$

(Here Prb abbreviates "provable." The subscript should perhaps be "A" instead of "T"; in constructing the sentence we utilize the recursiveness of the set A of axioms.)

> **LEMMA 37A**   Let T be a recursively axiomatizable theory as above.
>
> (a) Whenever $T \vdash \sigma$ then $A_E \vdash \mathrm{Prb}_T \, \sigma$.
> (b) If in addition T includes A_E, then T has the "reflection" property:
>
> $$T \vdash \sigma \implies T \vdash \mathrm{Prb}_T \, \sigma.$$

> PROOF.   If $T \vdash \sigma$ then we can let d be the number of a deduction of σ from the axioms A for T. We have $A_E \vdash \pi(\mathbf{S}^{\sharp\sigma}\mathbf{0}, \mathbf{S}^d\mathbf{0})$, and hence $A_E \vdash \mathrm{Prb}_T \, \sigma$. This gives part (a), from which part (b) follows immediately.   ⊣

Thus under modest assumptions, whenever T proves a sentence, it *knows* that it proves the sentence. Note that part (b) does *not* say that $T \vdash (\sigma \rightarrow \mathrm{Prb}_T \, \sigma)$. For example, if σ is true (in $\mathfrak{N}$) but unprovable from A_E, then the sentence $(\sigma \rightarrow \mathrm{Prb}_{A_E} \, \sigma)$ is *not* provable from A_E, and in fact is false in $\mathfrak{N}$.

Returning now to the proof of the Gödel incompleteness theorem (in the self-reference approach), we can apply the fixed-point lemma to obtain a sentence σ asserting its own unprovability in T:

$$A_E \vdash (\sigma \leftrightarrow \neg \mathrm{Prb}_T \, \sigma).$$

The following lemma provides part of the incompleteness theorem (the other part being Exercise 2 in Section 3.5):

LEMMA 37B Let T be a recursively axiomatizable theory including A_E and let σ be obtained from the fixed-point lemma as above. If T is consistent, then $T \nvdash \sigma$.

PROOF

$$
\begin{aligned}
T \vdash \sigma \;&\Rightarrow\; T \vdash \mathrm{Prb}_T \, \sigma \quad \text{by reflection} \\
&\Rightarrow\; T \vdash \neg \sigma \qquad\quad \text{by choice of } \sigma
\end{aligned}
$$

whence T is inconsistent. $\dashv$

So far, this lemma merely reflects ideas employed in Section 3.5, and the proof of Lemma 37B was not very complex. And that is exactly the point: The proof is *not* very complex, so perhaps it can be carried out *within* the theory T, if T is "sufficiently strong." That, we can hope that the steps

$$
\begin{aligned}
\mathrm{Prb}_T \, \sigma \;&\rightarrow\; \mathrm{Prb}_T \, \mathrm{Prb}_T \, \sigma \\
&\rightarrow\; \mathrm{Prb}_T \, \neg \sigma \\
&\rightarrow\; \mathrm{Prb}_T \, \mathbf{0} = \mathbf{S0}
\end{aligned}
$$

can be carried out in a sufficiently strong extension T of A_E.

If so, we get a remarkable conclusion. Let $\mathrm{Cons}\, T$ be the sentence $\neg\, \mathrm{Prb}_T \, \mathbf{0} = \mathbf{S0}$, which we think of as saying "T is consistent." (Here $\mathbf{0} = \mathbf{S0}$ is chosen simply as a convenient sentence refutable from A_E.) If T lets us carry out the steps in the preceding paragraph, then we can conclude:

$$
T \nvdash \mathrm{Cons}\, T, \quad \text{unless } T \text{ is } \textit{inconsistent}
$$

(Of course, an inconsistent theory contains every sentence, including sentences asserting — falsely — the theory's consistency. The situation we are finding here is that, under suitable assumptions, this is the *only* way that a theory can prove its own consistency.) Let's check the details: Suppose $T \vdash \mathrm{Cons}\, T$. Then by the preceding paragraph, $T \vdash \neg\, \mathrm{Prb}_T \, \sigma$. By choice of σ, we then get $T \vdash \sigma$. Lemma 37B then applies.

To make matters less vague, call the theory T *sufficiently strong* if it meets the following three "derivability" conditions.

1. $A_E \subseteq T$. This implies by Lemma 37A that T has the reflection property, $T \vdash \sigma \Rightarrow T \vdash \mathrm{Prb}_T \, \sigma$.

2. For any sentence σ, $T \vdash (\mathrm{Prb}_T \, \sigma \rightarrow \mathrm{Prb}_T \, \mathrm{Prb}_T \, \sigma)$. This is the reflection property, formalized within T.

3. For any sentences ρ and σ, $T \vdash (\mathrm{Prb}_T \, (\rho \rightarrow \sigma) \rightarrow (\mathrm{Prb}_T \, \rho \rightarrow \mathrm{Prb}_T \, \sigma))$. This is modus ponens, formalized within T.

FORMALIZED LEMMA 37B Assume that T is a sufficiently strong recursively axiomatizable theory, and let σ be a sentence such that

$$
A_E \vdash (\sigma \leftrightarrow \neg\, \mathrm{Prb}_T \, \sigma).
$$

Then $T \vdash (\mathrm{Cons}\, T \rightarrow \neg\, \mathrm{Prb}_T \, \sigma)$.

PROOF.   We put the pieces together carefully. By the choice of σ we get

$$T \vdash (\sigma \rightarrow (\mathrm{Prb}_T\, \sigma \rightarrow \mathbf{0} = \mathbf{S0})).$$

Applying first reflection and then formalized modus ponens to this formula yields

$$T \vdash (\mathrm{Prb}_T\, \sigma \rightarrow \mathrm{Prb}_T\, (\mathrm{Prb}_T\, \sigma \rightarrow \mathbf{0} = \mathbf{S0}))$$

after which another application of formalized modus ponens yields

$$T \vdash (\mathrm{Prb}_T\, \sigma \rightarrow (\mathrm{Prb}_T\, \mathrm{Prb}_T\, \sigma \rightarrow \neg\, \mathrm{Cons}\, T)).$$

The formula displayed above (to the right of the turnstile), together with $\mathrm{Prb}_T\, \sigma \rightarrow \mathrm{Prb}_T\, \mathrm{Prb}_T\, \sigma$ (formalized reflection) imply by sentential logic $\mathrm{Prb}_T\, \sigma \rightarrow \neg\, \mathrm{Cons}\, T$. ⊣

GÖDEL'S SECOND INCOMPLETENESS THEOREM (1931)   Assume that T is a sufficiently strong recursively axiomatizable theory. Then $T \vdash \mathrm{Cons}\, T$ if and only if T is *in*consistent.

PROOF.   If $T \vdash \mathrm{Cons}\, T$ then by Formalized Lemma 37B we have $T \vdash \neg\, \mathrm{Prb}_T\, \sigma$ whence by our choice of σ, we have $T \vdash \sigma$. We conclude from the (unformalized) Lemma 37B that T is inconsistent. ⊣

We can squeeze a bit more out of these ideas. Lemma 37B can be regarded as a special case (where τ is $\mathbf{0} = \mathbf{S0}$) of the following:

LEMMA 37C   Let T be a recursively axiomatizable theory including A_E, let τ be a sentence, and let σ be obtained from the fixed-point lemma so that

$$A_E \vdash (\sigma \leftrightarrow (\mathrm{Prb}_T\, \sigma \rightarrow \tau)).$$

If $T \vdash \sigma$, then $T \vdash \tau$.

PROOF.   We can think of σ as saying, "If I am provable, then τ." If $T \vdash \sigma$ then by reflection $T \vdash \mathrm{Prb}_T\, \sigma$. By the choice of σ, we have $T \vdash \tau$. ⊣

Actually we are not interested in this lemma, but in its formalization:

FORMALIZED LEMMA 37C   Assume that T is a sufficiently strong recursively axiomatizable theory. Let τ be a sentence, and let σ be a sentence such that

$$A_E \vdash (\sigma \leftrightarrow (\mathrm{Prb}_T\, \sigma \rightarrow \tau)).$$

Then $T \vdash (\mathrm{Prb}_T\, \sigma \rightarrow \mathrm{Prb}_T\, \tau)$.

PROOF.   We proceed as before. By the choice of σ we get

$$T \vdash (\sigma \rightarrow (\text{Prb}_T \, \sigma \rightarrow \tau)).$$

Applying first reflection and then formalized modus ponens to this formula yields

$$T \vdash (\text{Prb}_T \, \sigma \rightarrow \text{Prb}_T \, (\text{Prb}_T \, \sigma \rightarrow \tau))$$

after which another application of formalized modus ponens yields

$$T \vdash (\text{Prb}_T \, \sigma \rightarrow (\text{Prb}_T \, \text{Prb}_T \, \sigma \rightarrow \text{Prb}_T \, \tau)).$$

The formula displayed above (to the right of the turnstile), together with $\text{Prb}_T \, \sigma \rightarrow \text{Prb}_T \, \text{Prb}_T \, \sigma$ (formalized reflection) imply by sentential logic $\text{Prb}_T \, \sigma \rightarrow \text{Prb}_T \, \tau$. ⊣

LÖB'S THEOREM (1955)   Assume that T is a sufficiently strong recursively axiomatizable theory. If τ is any sentence for which $T \vdash (\text{Prb}_T \, \tau \rightarrow \tau)$, then $T \vdash \tau$.

Clearly if $T \vdash \tau$, then $T \vdash (\rho \rightarrow \tau)$ for any sentence ρ. So the conclusion to Löb's theorem can be stated

$$T \vdash (\text{Prb}_T \, \tau \rightarrow \tau) \iff T \vdash \tau.$$

PROOF.   Given the sentence τ, we construct σ to say, "If I am provable then τ," as above. Suppose that $T \vdash (\text{Prb}_T \, \tau \rightarrow \tau)$. By Formalized Lemma 37C we have $T \vdash (\text{Prb}_T \, \sigma \rightarrow \text{Prb}_T \, \tau)$. By our choice of σ, we conclude that $T \vdash \sigma$. So by the (unformalized) Lemma 37C, we have $T \vdash \tau$. ⊣

Löb's theorem was originally devised in order to solve the problem given in Exercise 1. But it implies (and in a sense is equivalent to) Gödel's second incompleteness theorem. Assume that T is a sufficiently strong axiomatizable theory. Applying Löb's theorem and taking τ to be $\mathbf{0} = \mathbf{S0}$, we have

$$T \vdash (\text{Prb}_T \, (\mathbf{0} = \mathbf{S0}) \rightarrow \mathbf{0} = \mathbf{S0}) \Rightarrow T \vdash \mathbf{0} = \mathbf{S0},$$

that is,

$$T \vdash \text{Cons } T \Rightarrow T \text{ is inconsistent.}$$

Thus we obtain a proof of the second incompleteness theorem.

But there is an issue not yet examined: What theories *are* sufficiently strong? Are there any at all (apart from the trivial case of the inconsistent theory)?

Yes, and here are two. The first is called "Peano arithmetic" (PA). Its axioms consist of the A_E axioms, plus all the "induction axioms." These are the universal closures of formulas having the form

$$\varphi(0) \wedge \forall x (\varphi(x) \rightarrow \varphi(\mathbf{S}x)) \rightarrow \forall x \, \varphi(x)$$

for a wff φ. The induction axioms — which state the ordinary principle of mathematical induction — enable us to carry out many arguments about the natural numbers (e.g., the commutative law of addition) within Peano arithmetic. But to be sure that formalized reflection and formalized modus ponens can be derived with Peano arithmetic, one must carry out the details, which we will not go through here.

We know that Peano arithmetic is consistent, because it is true in $\mathfrak{N}$. But by the second incompleteness theorem, PA cannot prove its own consistency. We "know" that PA is consistent by means of an argument we carry out either in informal mathematics, or — if we want — in set theory. So set theory has a higher "consistency strength" than PA: It proves the consistency of PA and PA does not.

A second sufficiently strong theory is axiomatic set theory. Or to be more careful, it is the set of sentences in the language of number theory that are provable in axiomatic set theory. The next subsection deals with this situation. This theory has the advantage that it is quite believable — on an informal level — that formalized reflection and formalized modus ponens are derivable. But what are our grounds for thinking that set theory is consistent? We know that PA is consistent because it is true in the "standard model" $\mathfrak{N}$ of number theory. It is not at all clear that we can meaningfully speak of a "standard model of set theory"!

Applications to Set Theory

We know that in the language of number theory, Cn A_E is incomplete and nonrecursive, as is any compatible recursively axiomatizable theory in the language.

But now suppose we leave arithmetic for a while and look at set theory. Here we have a language (with the parameters $\forall$ and $\in$) and a set of axioms. In all presently accepted cases the set of axioms is recursive. Or more precisely, the set of Gödel numbers of the axioms is recursive. And so the theory (set theory) obtained is recursively enumerable. We claim that this theory, if consistent, is not recursive and hence not complete. We can already sketch the argument in rough form. We can, in a very real sense, embed the language of number theory in set theory. We can then look at that fragment of set theory which deals with the natural numbers and their arithmetic (the shaded area in Fig. 14). That is a theory compatible with A_E. And so it is nonrecursive. Now if set theory were recursive, then its arithmetical part would also be recursive, which it is not. As a bonus, we will come across the second incompleteness theorem for the case of set theory.

Henceforth by set theory (ST) we mean that theory (in the language with equality having the two parameters $\forall$ and $\in$) which is the set of consequences of the reader's favorite set-theoretic axioms. (The standard Zermelo–Fraenkel axioms will do nicely, if the reader has no favorite.
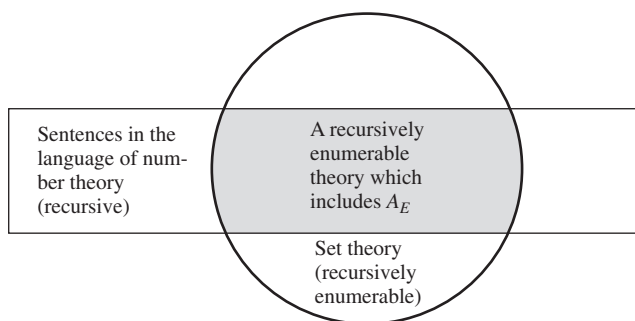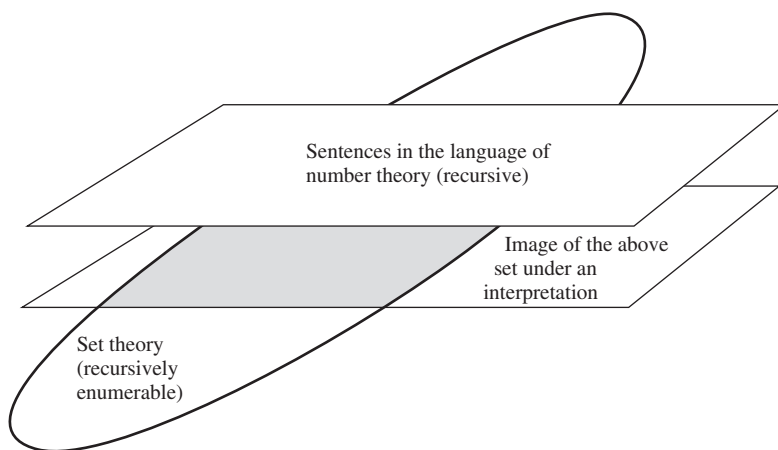
Figure 14. Set theory and number theory. (a) Flat picture. (b) A more accurate picture.

We ask only that the set of axioms be recursive, and that it be strong enough to yield certain everyday facts about sets.) We need an interpretation π of Cn A_E into ST. (The remainder of this section assumes a familiarity with Section 2.7.) But the existence of such a π is a standard result of set theory, although it is not usually stated in these words. We need formulas of the language of ST that adequately express the concept of being a natural number, being the sum of two given numbers, and so forth. To find these formulas, we turn to the way in which the arithmetic of natural numbers can be "embedded" in set theory. That is, on the one hand, natural numbers such as 2 or 7 do not *appear* to be sets. On the other hand, we can, when we choose, select sets to *represent* numbers. The standard approach is to take 0 to be the set $\varnothing$ and $n + 1$ to be the set $n; n$. This has the fringe benefit that each number is the set of all smaller numbers (e.g., $3 \in 7$). Let ω be the collection of all these sets

(these "number-sets"); thus ω is the set representing $\mathbb{N}$.

The formula $\pi_\forall$ is the result of eliminating the defined symbol ω from the formula $v_1 \in \omega$. The formula $\pi_\mathbf{0}$ is similarly obtained from the set-theoretic formula $v_1 = \varnothing$, and the formula $\pi_\mathbf{S}$ is obtained from $v_2 = v_1 \cup \{v_1\}$. The formula $\pi_<$ is simply $v_1 \in v_2$. For π_+ we use the translation into the language of ST of

> For any f,   if $f : \omega \times \omega \to \omega$ and for all a and b
> in ω we have $f(a, \varnothing) = a$
> and $f(a, b \cup \{b\}) = f(a, b) \cup \{f(a, b)\}$,
> then $f(v_1, v_2) = v_3$.

(The manner of translation is partially indicated in Chapter 0.) The formulas $\pi_\cdot$ and $\pi_\mathbf{E}$ are obtained in much the same fashion.

The claim that this π is an interpretation of Cn A_E into ST makes a number (and the number is 17) of demands on ST.

(i) $\exists v_1 \pi_\forall$ must be in ST. It is, since we can prove in set theory that ω is nonempty.

(ii) For each of the five function symbols f in the language of A_E, ST must contain a sentence asserting, roughly, that π_f defines a function on the set defined by $\pi_\forall$. (The exact sentence is set forth in the definition of interpretation in Section 2.7.) In the case of $\mathbf{0}$, we have in ST the result that there is a unique empty set and that it belongs to ω. The case for $\mathbf{S}$ is simple, since $\pi_\mathbf{S}$ defines a unary operation on the universe of all sets, and ω is closed under this operation. For $+$ we must use the recursion theorem on ω. That is, we can prove in ST (as sketched in Section 1.4) that there is a unique $f : \omega \times \omega \to \omega$ such that $f(a, \varnothing) = a$ and $f(a, b \cup \{b\}) = f(a, b) \cup \{f(a, b)\}$ for a, b in ω. The required property of π_+ then follows. Similar arguments apply to $\cdot$ and $\mathbf{E}$.

(iii) For each of the 11 sentences σ in A_E, the sentence σ^π must be in ST. For example, in the case of L3, we have in ST the fact that for any m and n in ω, either $m \in n$, $m = n$, or $n \in m$.

Since these demands are finite in number, there is a finite $\Phi \subseteq$ ST such that π is also an interpretation of Cn A_E into Cn Φ.

THEOREM 37D (STRONG UNDECIDABILITY OF SET THEORY)   Let T be a theory in the language of set theory such that $T \cup$ ST (or at least $T \cup \Phi$) is consistent. Then $\sharp T$ is not recursive.

PROOF.   Let Δ be the consistent theory $\mathrm{Cn}(T \cup \Phi)$. Let Δ_0 be the corresponding theory $\pi^{-1}[\Delta]$ in the language of number theory. From Section 2.7 we know that Δ_0 is a consistent theory (since Δ is). Also $A_E \subseteq \Delta_0$, since if $\sigma \in A_E$, then $\sigma^\pi \in \mathrm{Cn}\,\Phi \subseteq \Delta$.

Hence by the strong undecidability of Cn A_E (Theorem 35C), $\sharp\Delta_0$ is not recursive.

Now we must derive the nonrecursiveness of T from that of Δ_0. We have

$$\sigma \in \Delta_0 \quad \text{iff} \quad \sigma^\pi \in \Delta$$

and by the lemma below, $\sharp\sigma^\pi$ depends recursively on $\sharp\sigma$. That is, $\sharp\Delta_0 \leq_m \sharp\Delta$. Hence $\sharp\Delta$ cannot be recursive, lest $\sharp\Delta_0$ be. Similarly, we have

$$\tau \in \Delta \quad \text{iff} \quad (\varphi \rightarrow \tau) \in T,$$

where φ is the conjunction of the members of Φ. Since $\sharp(\varphi \rightarrow \tau)$ depends recursively on $\sharp\tau$, we have $\sharp\Delta \leq_m \sharp T$ so that $\sharp T$ cannot be recursive lest $\sharp\Delta$ be. ⊣

LEMMA 37E There is a recursive function p such that for any formula α of the language of number theory, $p(\sharp\alpha) = \sharp(\alpha^\pi)$.

PROOF. In Section 2.7 we gave explicit instructions for constructing α^π. The construction in some cases utilized formulas β^π for formulas β simpler than α. The methods of Sections 3.3 and 3.4 can be applied to the Gödel numbers of these formulas to show that p is recursive. But the details are not particularly attractive, and we omit them here. ⊣

COROLLARY 37F If set theory is consistent, then it is not complete.

PROOF. Set theory has a recursive set of axioms. If complete, the theory is then recursive (by item 21 of Section 3.4). By the foregoing theorem, this cannot happen if ST is consistent. ⊣

COROLLARY 37G In the language with equality and a two-place predicate symbol, the set of (Gödel numbers of) valid sentences is not recursive.

PARTIAL PROOF. In the foregoing theorem take $T = \text{Cn}\varnothing$, the set of valid sentences. The theorem then assures us that $\sharp T$ is nonrecursive, provided that Φ is consistent. We have not given the finite set Φ explicitly. But we assure the reader that Φ can be chosen in such a way as to be provably consistent. ⊣

It should be noted that π is *not* an interpretation of Th $\mathfrak{N}$ into ST (unless ST is inconsistent). For $\pi^{-1}[\text{ST}]$ is a recursively enumerable theory in the language of $\mathfrak{N}$, as a consequence of Lemma 37E. Hence it cannot coincide with Th $\mathfrak{N}$, and it can include the complete theory Th $\mathfrak{N}$ only if it is inconsistent.

Gödel's Second Incompleteness Theorem for Set Theory

We can employ our usual tricks to find a sentence a of number theory which indirectly asserts that its own interpretation σ^π is not a theorem of set theory. For let D be the ternary relation on $\mathbb{N}$ such that

$\langle a, b, c \rangle \in D$ iff a is the Gödel number of a formula α of number theory and c is the Gödel number of a deduction from the axioms of ST of $\alpha(\mathbf{S}^b\mathbf{0})^\pi$.

The relation D is recursive (by the usual arguments); let $\delta(v_1, v_2, v_3)$ represent D in Cn A_E. Let r be the Gödel number of

$$\forall v_3 \neg \delta(v_1, v_1, v_3)$$

and let σ be

$$\forall v_3 \neg \delta(\mathbf{S}^r\mathbf{0}, \mathbf{S}^r\mathbf{0}, v_3).$$

Observe that σ *does* indirectly assert that $\sigma^\pi \notin$ ST. We will now prove that the assertion is correct:

LEMMA 37H   If ST is consistent, then $\sigma^\pi \notin$ ST.

PROOF.   Suppose to the contrary that σ^π is deducible from the axioms of ST; let k be $\mathcal{G}$ of such a deduction. Then $\langle r, r, k \rangle \in D$.

$$\therefore A_E \vdash \delta(\mathbf{S}^r\mathbf{0}, \mathbf{S}^r\mathbf{0}, \mathbf{S}^k\mathbf{0});$$
$$\therefore A_E \vdash \exists v_3 \delta(\mathbf{S}^r\mathbf{0}, \mathbf{S}^r\mathbf{0}, v_3);$$

i.e.,

$$A_E \vdash \neg\sigma.$$

Applying our interpretation π, we conclude that $\neg\sigma^\pi$ is in ST, whence ST is inconsistent. Thus

$$\text{ST is consistent} \Rightarrow \sigma^\pi \notin \text{ST}. \qquad\dashv$$

Now the above proof, like all those in this book, is carried out in informal mathematics. But all of our work in the book could have been carried out within ST. Indeed it is common knowledge that essentially all work in mathematics can be carried out in ST. Imagine actually doing so. Then instead of a proof of an English sentence, "ST is consistent $\Rightarrow \sigma^\pi \notin$ ST," we have a deduction from the axioms of ST of a certain sentence in the formal language of set theory:

$$(\text{Cons}(\text{ST}) \to \square).$$

Here Cons(ST) is the result of translating (in a nice way) "ST is consistent" into the language of set theory. Similarly, $\square$ is the result of translating "$\sigma^\pi \notin$ ST." But we already *have* a sentence in the language

of set theory asserting that $\sigma^\pi \notin$ ST. It is σ^π. This strongly suggests that $\square$ is (or is provably equivalent in ST to) σ^π, from which we get

$$(\mathrm{Cons}(\mathrm{ST}) \rightarrow \sigma^\pi)$$

as a theorem of ST.

Now this *can* actually be carried out in such a way as to have $\square$ be σ^π. We have given above an argument, which we hope will convince the reader that this is at least probable. And from it we now have the result:

> **GÖDEL'S SECOND INCOMPLETENESS THEOREM FOR SET THEORY**   The sentence Cons(ST) is not a theorem of ST, unless ST is inconsistent.

> PROOF.   By the above (plausibility) argument
>
> $$(\mathrm{Cons}(\mathrm{ST}) \rightarrow \sigma^\pi)$$
>
> is a theorem of ST. So if Cons(ST) is also a theorem of ST, then σ^π is, too. But by Lemma 37H, if $\sigma^\pi \in$ ST, then ST is inconsistent.
> $\dashv$

Of course if ST is inconsistent, then every sentence is a theorem, including Cons(ST). Because of this, a proof of Cons(ST) within ST would not convince people that ST was consistent. (And by Gödel's second theorem, it would convince them of the opposite.) But prior to Gödel's work it was possible to hope that Cons(ST) might be provable from assumptions weaker than the axioms of set theory, ideally assumptions already known to be consistent. But we now see that Cons(ST) is not in any subtheory of ST, unless of course ST is inconsistent.

We are left with the conclusion that any recursively axiomatizable theory of sets (provided it meets the desirable conditions of being consistent and strong enough to prove everyday facts) is an incomplete theory. This raises a challenge: to find additional axioms to add to the theory. On the one hand, we want the additional axioms to strengthen the theory in useful ways. On the other hand, we want the additional axioms to reflect accurately our informal ideas about what sets really are and how they really behave.

Exercises

1. Let σ be a sentence such that

$$A_E \vdash (\sigma \leftrightarrow \mathrm{Prb}_{A_E}\sigma).$$

(Thus σ says "I am provable," in contrast to the sentence "I am unprovable" that has been found to have such interesting properties.) Does $A_E \vdash \sigma$?

2. Let T be a theory in a recursively numbered language, and assume that there is an interpretation of Cn A_E into T. Show that T is strongly undecidable; i.e., whenever T' is a theory in the language for which $T \cup T'$ is consistent, then $\sharp T'$ is not recursive.

SECTION 3.8

Representing Exponentiation[1]

In Sections 3.1 and 3.2 we studied the theory of certain reducts of $\mathfrak{N}$ and found them to be decidable. Then in Section 3.3 we added *both* multiplication and exponentiation. The resulting theory was found (in Section 3.5) to be undecidable. Actually it would have been enough to add only multiplication (and forego exponentiation); we would still have undecidability.

Let $\mathfrak{N}_M$ be the reduct of $\mathfrak{N}$ obtained by dropping exponentiation:

$$\mathfrak{N}_M = (\mathbb{N}; 0, S, <, +, \cdot).$$

Thus the symbol $\mathbf{E}$ does not appear in the language of $\mathfrak{N}_M$. Let A_M be the set obtained from A_E by dropping E1 and E2. The purpose of this section is to show that all the theorems of Sections 3.3–3.5 continue to hold when "A_E" and "$\mathfrak{N}$" are replaced by "A_M" and "$\mathfrak{N}_M$." The key fact needed to establish this claim is that exponentiation is representable in Cn A_M. That is, there is a formula ε in the language of $\mathfrak{N}_M$ such that for any a and b,

$$A_M \vdash \forall z[\varepsilon(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}, z) \leftrightarrow z = \mathbf{S}^{(a^b)}\mathbf{0}].$$

Thus $\varepsilon(x, y, z)$ can be used to simulate the formula $x\mathbf{E}y = z$ without actual use of the symbol $\mathbf{E}$.

If we look to see what relations and functions are representable in Cn A_M, we find at first that everything (except for exponentiation itself) that was shown to be representable in Cn A_E is (by the same proof) representable in Cn A_M. Until, that is, we reach item 7 in the catalog listing of Section 3.3. To go further, we must show that exponentiation itself is representable in Cn A_M.

We know that exponentiation can be characterized by the recursion equations

$$a^0 = 1,$$
$$a^{b+1} = a^b \cdot a.$$

---

[1] This section may be omitted without loss of continuity.

From what we know about primitive recursion (catalog item 13 in Section 3.3 plus Exercise 8 there), we might think of defining

$$E^*(a, b) = \text{the least } s \text{ such that } [(s)_0 = 1 \text{ and}$$
$$\text{for all } i < b, \ (s)_{i+1} = (s)_i \cdot a].$$

For then $a^b = (E^*(a, b))_b$. This fails to yield a proof of representability, because we do not yet know that the decomposition function $(a)_b$ is representable in $\text{Cn } A_M$. But we do not really need that particular decomposition function (which corresponded to a particular way of encoding sequences). All we need is *some* function δ that acts like a decomposition function; the properties we need are summarized in the following lemma.

> **LEMMA 38A**   There is a function δ representable in $\text{Cn } A_M$ such that for every $n, a_0, \ldots, a_n$, there is an s for which $\delta(s, i) = a_i$ for all $i \leq n$.

Once the lemma has been established, we can define

$$E^{**}(a, b) = \text{the least } s \text{ such that } [\delta(s, 0) = 1 \text{ and}$$
$$\text{for all } i < b, \ \delta(s, i + 1) = \delta(s, i) \cdot a].$$

The lemma assures us that such an s exists. E^{**} is then representable in $\text{Cn } A_M$, as is exponentiation, since

$$a^b = \delta(E^{**}(a, b), b).$$

A function δ that establishes the lemma will be provided by some facts of number theory.

A Pairing Function

As a first step toward proving the foregoing lemma, we will construct functions for encoding and decoding pairs of numbers. It is well known that there exist functions mapping $\mathbb{N} \times \mathbb{N}$ one-to-one onto $\mathbb{N}$. In particular, the function J does this, where in the diagram shown, $J(a, b)$ has been written at the point with coordinates $\langle a, b \rangle$.
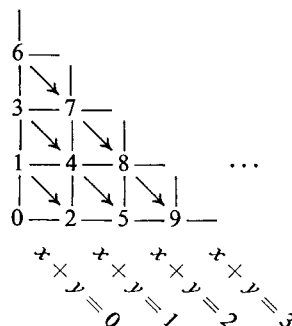
For example, $J(2, 1) = 8$ and $J(0, 2) = 3$. To obtain an equation for $J(a, b)$, we note that along the line $x + y = n$ there are $n + 1$ points (with coordinates in $\mathbb{N}$). Thus

$J(a, b) =$ the number of points in the plane to which J assigns smaller
       values
    $=$ [the number of points on lines $x + y = n$ for
      $n = 0, 1, \ldots, (a + b - 1)$] + [the number of points on the
      line $x + y = a + b$ for which $x < a$]
    $= [1 + 2 + \cdots + (a + b)] + a$
    $= \frac{1}{2}(a + b)(a + b + 1) + a$
    $= \frac{1}{2}[(a + b)^2 + 3a + b].$

Let K and L be the corresponding projection functions onto the axes, i.e., the unique functions such that

$$K(J(a, b)) = a, \quad L(J(a, b)) = b.$$

For example, $K(7) = 1$, the x-coordinate of the point $\langle 1, 2 \rangle$ in the plane to which J assigned the number 7. Similarly, $L(7) = 2$, the y-coordinate of that point.

We claim that J, K, and L are representable in Cn A_M. The function

$$H(a) = \text{the least } b \text{ such that } a \leq 2b$$

has the property that $H(a) = \frac{1}{2}a$ for even a. Then we can write

$J(a, b) = H((a + b) \cdot (a + b + 1)) + a,$
  $K(p) =$ the least a such that [for some $b \leq p$, $J(a, b) = p$],
  $L(p) =$ the least b such that [for some $a \leq p$, $J(a, b) = p$].

From the form of the four preceding equations we conclude that H, J, K, and L are representable in Cn A_M.

The Gödel β-function

Let β be the function defined as follows:

$\beta(c, d, i) =$ the remainder in $c \div [1 + (i + 1) \cdot d]$
        $=$ the least r such that for some $q \leq c$,
        $c = q \cdot [1 + (i + 1) \cdot d] + r.$

This unlikely-looking function produces a satisfactory decomposition function for Lemma 38A. Let

$$\delta(s, i) = \beta(K(s), L(s), i).$$

It is clear that δ is representable in Cn A_M. What is not so obvious is that it meets the conditions of Lemma 38A. We want to show:

      For any n and any $a_0, \ldots, a_n$, there are numbers
      c and d such that $\beta(c, d, i) = a_i$ for all $i \leq n$.   (*)

For then it follows that $\delta(J(c, d), i) = \beta(c, d, i) = a_i$ for $i \leq n$.

Now ($*$) is a statement of number theory, not logic. The proof of ($*$) is based on the Chinese remainder theorem. Numbers $d_0, \ldots, d_n$ are said to be *relatively prime in pairs* iff no prime divides both d_i and d_j for $i \neq j$.

CHINESE REMAINDER THEOREM   Let $d_0, \ldots, d_n$, be relatively prime in pairs; let $a_0, \ldots, a_n$ be natural numbers with each $a_i < d_i$. Then we can find a number c such that for all $i \leq n$,

$$a_i = \text{the remainder in } c \div d_i.$$

PROOF.   Let $p = \Pi_{i \leq n} d_i$, and for any c let $F(c)$ be the $(n+1)$-tuple of remainders when c is divided by $d_0, \ldots, d_n$. Notice that there are p possible values for this $(n + 1)$-tuple.

We claim that F is one-to-one on $\{k \mid 0 \leq k < p\}$. For suppose that $F(c_1) = F(c_2)$. Then each d_i divides $|c_1 - c_2|$. Since the d_i's are relatively prime, p must divide $|c_1 - c_2|$. For c_1, c_2 less than p, this implies that $c_1 = c_2$.

Hence the restriction of F to $\{k \mid 0 \leq k < p\}$ takes on all p possible values. In particular, it assumes (at some point c) the value $\langle a_0, \ldots, a_n \rangle$. And that is the c we want. $\dashv$

LEMMA 38B   For any $s \geq 0$, the $s + 1$ numbers

$$1 + 1 \cdot s!, \quad 1 + 2 \cdot s!, \quad \ldots, \quad 1 + (s + 1) \cdot s!$$

are relatively prime in pairs.

PROOF.   All these numbers have the property that any prime factor q cannot divide $s!$, whence $q > s$. If the prime q divides both $1 + j \cdot s!$ and $1 + k \cdot s!$, then it divides their difference, $|j - k| \cdot s!$. Since q does not divide $s!$, it divides $|j - k|$. But $|j - k| \leq s < q$. This is possible only if $|j - k| = 0$. $\dashv$

PROOF OF ($*$).   Assume we are given $a_0, \ldots, a_n$; we need numbers c and d such that the remainder when c is divided by $1 + (i + 1) \cdot d$ is a_i, for $i \leq n$.

Let s be the largest of $\{n, a_0, \ldots, a_n\}$ and let $d = s!$. Then by Lemma 38B, the numbers $1 + (i + 1) \cdot d$ are relatively prime in pairs for $i \leq n$. So by the Chinese remainder theorem there is a c such that the remainder in $c \div [1 + (i + 1) \cdot d]$ is a_i for $i \leq n$. $\dashv$

This completes the proof of Lemma 38A. And by the argument that followed that lemma, we can conclude:

THEOREM 38C   Exponentiation is representable in Cn A_M.

Armed with this theorem, we can now return to catalog item 7 of Section 3.3. The proof given there now establishes that the function in question (whose value at n is p_n) is representable in Cn A_M. For it was

TABLE X

Structure	Theory	Models of the theory	Definable sets	Comments
$(\mathbb{N})$	Decidable. Not finitely axiomatizable. Admits elimination of quantifiers.	Any infinite set.	$\varnothing$ and $\mathbb{N}$. $\{0\}$ is not definable.	
$(\mathbb{N}; 0)$	As above.	Any infinite set with distinguished element.	$\varnothing, \{0\}, \mathbb{N} - \{0\}, \mathbb{N}$. S is not definable.	
$(\mathbb{N}; 0, S)$	As above.	Standard part plus any number of Z-chains.	Finite and cofinite sets. $<$ is not definable.	$\{0\}$ is definable in $(\mathbb{N}; S)$.
$(\mathbb{N}; 0, S, <)$	Decidable. Finitely axiomatizable. Admits elimination of quantifiers.	As above, with any ordering of the Z-chains.	Finite and cofinite sets. $+$ is not definable.	$\{0\}$ and S are definable in $(\mathbb{N}; <)$.
$(\mathbb{N}; 0, S, <, +)$	Decidable (Presburger).	The Z-chains are densely ordered without endpoints. Also there is a suitable addition operation.	Eventually periodic sets. $\cdot$ is not definable.	$\{0\}$, S, and $<$ are definable in $(\mathbb{N}; +)$.
$(\mathbb{N}; 0, S, <, +, \cdot)$	Not arithmetical. $\therefore$ not recursively axiomatizable.	As above, but with a suitable multiplication operation.	All arithmetical relations are definable.	The arithmetical relations are definable in $(\mathbb{N}; S, \cdot), (\mathbb{N}; +, \cdot)$, and $(\mathbb{N}; <, D)$, where $D(x, y) = (x)_y$.

formed by allowable methods from relations and functions (including exponentiation) known to be representable in $\text{Cn } A_M$.

The same phenomenon persists throughout Sections 3.3 and 3.4. The representability proofs given there now establish representability in $\text{Cn } A_M$. Thus any recursive relation is representable in $\text{Cn } A_M$, and if the relation happens to be a function, then it is functionally representable. The proofs given in Section 3.5 then apply to $\mathfrak{N}_M$ and A_M as well as to $\mathfrak{N}$ and A_E. In particular, we have the strong undecidability of $\text{Cn } A_M$: Any theory T in the language of $\mathfrak{N}_M$ for which $T \cup A_M$ is consistent cannot be recursive.

Notice that any relation definable in $\mathfrak{N}$ (i.e., any arithmetical relation) is also definable in $\mathfrak{N}_M$. For exponentiation, being representable in a subtheory of $\text{Th } \mathfrak{N}_M$, is *a fortiori* definable in $\mathfrak{N}_M$. By the new version of Tarski's theorem, $\sharp \text{Th} \mathfrak{N}_M$ is not definable in $\mathfrak{N}_M$, and consequently $\sharp \text{Th} \mathfrak{N}_M$ cannot be arithmetical.

In the terminology of Section 2.7, we can say that there is a faithful interpretation of $\text{Th } \mathfrak{N}$ into $\text{Th } \mathfrak{N}_M$. It equals the identity interpretation on all parameters except **E**, and to **E** it assigns a formula defining exponentiation in $\mathfrak{N}_M$.

In Table X we summarize some of the results of Chapter 3 on number theory and its reducts.

Exercises

1. Let $D(a, b) = (a)_b$. Show that any arithmetical relation is definable in the structure $(\mathbb{N}; <, D)$. *Remark*: One may well ask why $\text{Th } \mathfrak{N}_A$, arithmetic with addition, is decidable (as shown in Section 3.2), while $\text{Th } \mathfrak{N}_M$, the theory of arithmetic with addition and multiplication, is undecidable. One answer is that, as this section shows, multiplication lets us do a certain amount of sequence coding and decoding. The point of this exercise is to show that once we have the decoding function D and ordering, we have the full complexity of arithmetic with addition, multiplication, and exponentiation.

2. Show that the addition relation $\{\langle a, b, c \rangle \mid a + b = c\}$ is definable in the structure $(\mathbb{N}; S, \cdot)$. *Suggestion*: Under what conditions does the equation $S(ac) \cdot S(bc) = S(c \cdot c \cdot S(ab))$ hold?

3. (a) Show that $\text{Th}(\mathbb{Z}; +, \cdot)$ is strongly undecidable. (See Exercise 2 of Section 3.7.)

   (b) (This part assumes a background in algebra.) Show that the theory of rings is undecidable and that the theory of commutative rings is undecidable.

Chapter
F O U R

Second-Order Logic

SECTION 4.1
Second-Order Languages

We can obtain — at a cost — richer, more expressive languages than the first-order languages considered thus far, by allowing quantification of predicate or function symbols. For example,

$$\exists x(Px \rightarrow \forall x\ Px)$$

is a valid formula having $\forall$ and P as its parameters. Because it is true no matter how P is interpreted,

$$\forall P\ \exists x(Px \rightarrow \forall x\ Px)$$

deserves to be called valid. (Now $\forall$ is the only parameter, since P is treated as a predicate variable.)

Suppose, then, that we have in addition to the symbols introduced at the beginning of Section 2.1, the further logical symbols:

4. Predicate variables: For each positive integer n we have the n-place predicate variables

$$X_1^n, X_2^n, \ldots.$$

5. Function variables: For each positive integer n, we have the n-place function variables

$$F_1^n, F_2^n, \ldots.$$

The usual variables $v_1, v_2, \ldots$ will now be called *individual* variables, to avoid confusion. The terms are as before defined as the expressions that can be built up from the constant symbols and the

individual variables by applying the function symbols (both the function parameters and the function variables). Atomic formulas are again expressions $Pt_1 \cdots t_n$, where $t_1, \ldots, t_n$ are terms and P is an n-place predicate symbol (parameter or variable). The definition of wff is augmented by new formula-building operations: If φ is a wff, then so also are $\forall X_i^n \varphi$ and $\forall F_i^n \varphi$. The concept of a variable occurring free in φ is defined just as before. A sentence is a wff σ in which no variable (individual, predicate, or function) occurs free.

It should be remarked that the roles played by predicate parameters and free predicate variables are essentially the same. There is the same close relationship between constant symbols and free individual variables, and between function parameters and free function variables.

By a *structure* we continue to mean a function on the set of parameters meeting the conditions set forth in Section 2.2. We must extend the definition of satisfaction in the natural way. Let V now be the set of all variables, individual, predicate, or function. Let s be a function on V that assigns to each variable the suitable type of object. Thus $s(v_1)$ is a member of the universe, $s(X^n)$ is an n-ary relation on the universe, and $s(F^n)$ is an n-ary operation. For a term t, $\bar{s}(t)$ is defined in the natural way. In particular, if F is a function variable, then $\bar{s}(Ft_1 \cdots t_n)$ is the result of applying the function $s(F)$ to $\langle \bar{s}(t_1), \ldots, \bar{s}(t_n) \rangle$. Satisfaction of atomic formulas is also defined essentially as before. For a predicate variable X,

$$\models_{\mathfrak{A}} Xt_1 \cdots t_n[s] \quad \text{iff} \quad \langle \bar{s}(t_1), \ldots, \bar{s}(t_n) \rangle \in s(X).$$

The only new features in the definition of satisfaction arise from our new quantifiers.

5. $\models_{\mathfrak{A}} \forall X_i^n \varphi[s]$ iff for every n-ary relation R on $|\mathfrak{A}|$, we have $\models_{\mathfrak{A}} \varphi[s(X_i^n \mid R)]$.

6. $\models_{\mathfrak{A}} \forall F_i^n \varphi[s]$ iff for every function $f : |\mathfrak{A}|^n \to |\mathfrak{A}|$, we have $\models_{\mathfrak{A}} \varphi[s(F_i^n \mid f)]$.

Again it is easy to see that only the values of s at variables occurring free in the formula are significant. For a sentence σ, we may unambiguously speak of its being true or false in $\mathfrak{A}$. Logical (semantical) implication is defined exactly as before.

EXAMPLE 1.   A well-ordering is an ordering relation such that any nonempty set has a least (with respect to the ordering) element. This last condition can be translated into the second-order sentence

$$\forall X(\exists y\, Xy \to \exists y(Xy \land \forall z(Xz \to y \leq z))).$$

Here, as elsewhere, we omit the subscripts on X and F when they are immaterial, and we omit the superscripts if they are clear from the context.

EXAMPLE 2. One of Peano's postulates (the induction postulate) states that any set of natural numbers that contains 0 and is closed under the successor function is, in fact, the set of all natural numbers. This can be translated into the second-order language for number theory as

$$\forall X(X\mathbf{0} \wedge \forall y(Xy \rightarrow X\mathbf{S}y) \rightarrow \forall y\, Xy).$$

Any model of S1, S2, and the above Peano induction postulate is isomorphic to $(\mathbb{N}; 0, S)$; see Exercise 1. Thus this set of sentences is categorical; i.e., all its models are isomorphic.

EXAMPLE 3. For any formula φ in which the predicate variable X^n does not occur free, the formula

$$\exists X^n\, \forall v_1 \cdots \forall v_n[X^n v_1 \cdots v_n \leftrightarrow \varphi]$$

is valid. (Here other variables may occur free in φ in addition to $v_1, \ldots, v_n$.) It says that there exists a relation consisting of exactly the n-tuples satisfying φ. Formulas of this form are called *relation comprehension formulas*. There are also the analogous *function comprehension formulas*. If ψ is a formula in which the variable F^n does not occur free, then

$$\forall v_1 \cdots \forall v_n\, \exists! v_{n+1}\, \psi \rightarrow$$
$$\exists F^n\, \forall v_1 \cdots \forall v_{n+1}(F^n v_1 \cdots v_n = v_{n+1} \leftrightarrow \psi)$$

is valid. (Here "$\exists! v_{n+1}\, \psi$" is an abbreviation for a formula obtained from Exercise 21 of Section 2.2.)

EXAMPLE 4. In the ordered field of real numbers, any bounded nonempty set has a least upper bound. We can translate this by the second-order sentence

$$\forall X[\exists y\, \forall z(Xz \rightarrow z \leq y) \wedge \exists z\, Xz \rightarrow$$
$$\exists y\, \forall y'(\forall z(Xz \rightarrow z \leq y') \leftrightarrow y \leq y')].$$

It is known that any ordered field that satisfies this second-order sentence is isomorphic to the ordered field of reals.

EXAMPLE 5. For each $n \geq 2$, we have a first-order sentence λ_n which translates, "There are at least n things." For example, λ_3 is

$$\exists x\, \exists y\, \exists z(x \neq y \wedge x \neq z \wedge y \neq z).$$

The set $\{\lambda_2, \lambda_3, \ldots\}$ has for its class of models the EC_Δ class consisting of the infinite structures. There is *a single* second-order sentence that is equivalent. A set is infinite iff there is an ordering on it having no last element. Or more simply, a set is

infinite iff there is a transitive irreflexive relation R on the set whose domain is the entire set. This condition can be translated into a second-order sentence λ_∞:

$$\exists X[\forall u \,\forall v \,\forall w (Xuv \rightarrow Xvw \rightarrow Xuw) \wedge \forall u \,\neg Xuu \,\wedge$$
$$\forall u \,\exists v \, Xuv].$$

Another sentence (using a function variable) that defines the class of infinite structures is

$$\exists F[\forall x \,\forall y (Fx = Fy \rightarrow x = y) \wedge \exists z \,\forall x \, Fx \neq z],$$

which says there is a one-to-one function that is not onto.

The preceding example shows that the compactness theorem fails for second-order logic:

THEOREM 41A   There is an unsatisfiable set of second-order sentences every finite subset of which is satisfiable.

PROOF.   The set is, in the notation of the above example,

$$\{\neg \lambda_\infty, \lambda_2, \lambda_3, \ldots\}. \hspace{3em} \dashv$$

The Löwenheim–Skolem theorem also fails for second-order logic. By the *language of equality* we mean the language (with $=$) having no parameters other than $\forall$. A structure for this language can be viewed as being simply a nonempty set. In particular, a structure is determined to within isomorphism by its cardinality. A sentence in this language is therefore determined to within logical equivalence by the set of cardinalities of its models (called its *spectrum*).

THEOREM 41B   There is a sentence in the second-order language of equality that is true in a set iff its cardinality is $2^{\aleph_0}$.

PROOF, USING CONCEPTS FROM ALGEBRA AND ANALYSIS.   Consider first the conjunction of the (first-order) axioms for an ordered field, further conjoined with the second-order sentence expressing the least-upper-bound property (see Example 4 of this section). This is a sentence whose models are exactly the isomorphs of the real ordered field (i.e., the structures isomorphic to the ordered field of real numbers). We now convert the parameters $\mathbf{0}$, $\mathbf{1}$, $+$, $\cdot$, $<$ to variables (individual, function, or predicate as appropriate) which we existentially quantify. The resulting sentence has the desired properties. $\hspace{2em} \dashv$

There are other cardinal numbers with second-order characterizations of this sort; cf. Exercise 2.

THEOREM 41C   The set of Gödel numbers of valid second-order sentences is not definable in $\mathfrak{N}$ by any second-order formula.

Here we assume that Gödel numbers have been assigned to second-order expressions in a manner like that used before. Although our proof applies to the second-order language of number theory, the theorem is true for any recursively numbered language having at least a two-place predicate symbol.

PROOF.   Let T^2 be the second-order theory of $\mathfrak{N}$, i.e., the set of second-order sentences true in $\mathfrak{N}$. The same argument used to prove Tarski's theorem shows that $\sharp T^2$ is not definable in $\mathfrak{N}$ by any second-order formula.

Now let α be the conjunction of the members of A_E with the second-order Peano induction postulate (Example 2). Any model of α is isomorphic to $\mathfrak{N}$; cf. Exercise 1. Consequently, for any sentence σ,

$$\sigma \in T^2 \quad \text{iff} \quad (\alpha \longrightarrow \sigma) \text{ is valid.}$$

Consequently, the set of (Gödel numbers of) validities cannot be definable lest $\sharp T^2$ be.   ⊣

A fortiori, the set of Gödel numbers of second-order validities is not arithmetical and not recursively enumerable. That is, the enumerability theorem fails for second-order logic. (In the other direction, one can show that this set is not definable in number theory of order three, or even of order ω. But these are topics we will not enter into here.)

It is interesting to compare the effect of a second-order universal sentence, such as the Peano induction postulate

$$\forall X(X\mathbf{0} \wedge \forall y(Xy \longrightarrow X\mathbf{S}y) \longrightarrow \forall y\, Xy)$$

and the corresponding first-order "schema," i.e., the set of all sentences

$$\varphi(\mathbf{0}) \wedge \forall y(\varphi(y) \longrightarrow \varphi(\mathbf{S}y)) \longrightarrow \forall y\, \varphi(y),$$

where φ is a first-order formula having just v_1 free. If $\mathfrak{A}$ is a model of the Peano induction postulate, then any subset of $|\mathfrak{A}|$ containing $\mathbf{0}^{\mathfrak{A}}$ and closed under $\mathbf{S}^{\mathfrak{A}}$ is in fact all of $|\mathfrak{A}|$. On the other hand, if $\mathfrak{A}$ is a model of the corresponding axiom schema, we can say only that every *definable* subset of $|\mathfrak{A}|$ containing $\mathbf{0}^{\mathfrak{A}}$ and closed under $\mathbf{S}^{\mathfrak{A}}$ is all of $|\mathfrak{A}|$. There may well be undefinable subsets for which this fails. (For example, take any model $\mathfrak{A}$ of $\mathrm{Th}(\mathbb{N}; 0, S)$ having Z-chains. Then $\mathfrak{A}$ satisfies the above first-order schema, but it does not satisfy the second-order induction postulate. The set of standard points is simply not definable in $\mathfrak{N}$.)

Exercises

1. Show that any structure for the language with parameters $\forall$, **0**, and **S** that satisfies the sentences

$$\forall x \, \mathbf{S}x \neq \mathbf{0} \tag{S1}$$

$$\forall X \forall y (\mathbf{S}x = \mathbf{S}y \rightarrow x = y) \tag{S2}$$

   and the Peano induction postulate

$$\forall X (X\mathbf{0} \wedge \forall y (Xy \rightarrow X\mathbf{S}y) \rightarrow \forall y \, Xy)$$

   is isomorphic to $\mathfrak{N}_S = (\mathbb{N}; 0, S)$.

2. (a) Give a sentence in the second-order language of equality that is true in a set iff its cardinality is $\aleph_0$.
   (b) Do the same for $\aleph_1$.

3. Let φ be a formula in which only the n-place predicate variable X occurs free. Say that an n-ary relation R on $|\mathfrak{A}|$ is *implicitly defined* in $\mathfrak{A}$ by φ iff $\mathfrak{A}$ satisfies φ with an assignment of R to X but does not satisfy φ with an assignment of any other relation to X. Show that $\sharp \operatorname{Th} \mathfrak{N}$, the set of Gödel numbers of first-order sentences true in $\mathfrak{N}$, is implicitly definable in $\mathfrak{N}$ by a formula without quantified predicate or function variables. *Suggestion*: The idea is to write down conditions that the set of true sentences must meet.

4. Consider a language (with equality) having the one-place predicate symbols I and S and the two-place predicate symbol E. Find a second-order sentence σ such that (i) if A is a set for which $A \cap \mathcal{P}A = \varnothing$ and if $|\mathfrak{A}| = A \cup \mathcal{P}A$, $I^{\mathfrak{A}} = A$, $S^{\mathfrak{A}} = \mathcal{P}A$, $E^{\mathfrak{A}} = \{\langle a, b \rangle \mid a \in b \subseteq A\}$, then $\mathfrak{A}$ is a model of σ; and (ii) every model of σ is isomorphic to one of the sort described in (i). *Remark*: Roughly speaking, σ translates "$S = \mathcal{P}I$."

SECTION 4.2
Skolem Functions

We want to show how, given any *first*-order formula, one can find a logically equivalent prenex second-order formula of a very special form:

existential quantifiers	universal individual quantifiers	quantifier-free formula

This is a prenex formula wherein all universal quantifiers are individual ones that follow a string of existential individual and function quantifiers.

In the simplest example, observe that

$$\forall x \, \exists y \, \varphi(x, y) \models\!\!=\!\!\models \exists F \, \forall x \, \varphi(x, Fx).$$

In the "$\models$" direction this is easy to see. For the "$\models$" direction, consider a structure $\mathfrak{A}$ and an assignment function s satisfying $\forall x \, \exists y \, \varphi(x, y)$. We know that for any $a \in |\mathfrak{A}|$ there is at least one $b \in |\mathfrak{A}|$ such that

$$\models_{\mathfrak{A}} \varphi(x, y)[s(x \mid a)(y \mid b)].$$

We obtain a function f on $|\mathfrak{A}|$ by choosing one such b for each a and taking $f(a) = b$. (The axiom of choice is used here.) Then

$$\models_{\mathfrak{A}} \forall x \, \varphi(x, Fx)[s(F \mid f)].$$

This function f is called a *Skolem function* for the formula $\forall x \, \exists y \, \varphi$ in the structure $\mathfrak{A}$.

The same argument applies more generally. As a second example, suppose that we begin with the formula

$$\exists y_1 \, \forall x_1 \, \exists y_2 \, \forall x_2 \, \forall x_3 \, \exists y_3 \, \psi(y_1, y_2, y_3).$$

(We have listed only y_1, y_2, and y_3, but possibly other variables occur free in ψ as well.) Here we already have the existential quantifier $\exists y_1$ at the left. What remains is

$$\forall x_1 \, \exists y_2 \, \forall x_2 \, \forall x_3 \, \exists y_3 \, \psi(y_1, y_2, y_3).$$

This is a special case of the first example (with $\varphi(x_1, y_2) = \forall x_2 \, \forall x_3 \, \exists y_3 \, \psi(y_1, y_2, y_3)$). It is logically equivalent, as before, to

$$\exists F_2 \, \forall x_1 \, \forall x_2 \, \forall x_3 \, \exists y_3 \, \psi(y_1, F_2 x_1, y_3).$$

Now we have the existential quantifiers $\exists y_1 \, \exists F_2$ at the left; what remains is

$$\forall x_1 \, \forall x_2 \, \forall x_3 \, \exists y_3 \, \psi(y_1, F_2 x_1, y_3).$$

By the same reasoning as before, this is logically equivalent to

$$\exists F_3 \, \forall x_1 \, \forall x_2 \, \forall x_3 \, \psi(y_1, F_2 x_1, F_3 x_1 x_2 x_3),$$

where F_3 is a three-place function variable. Thus the original formula is equivalent to

$$\exists y_1 \, \exists F_2 \, \exists F_3 \, \forall x_1 \, \forall x_2 \, \forall x_3 \, \psi(y_1, F_2 x_1, F_3 x_1 x_2 x_3).$$

For quantifier-free ψ, this is in the form we desire.

SKOLEM NORMAL FORM THEOREM   For any first-order formula, we can find a logically equivalent second-order formula consisting of:

(a) First a string (possibly empty) of existential individual and function quantifiers, followed by
(b) A string (possibly empty) of universal individual quantifiers, followed by
(c) A quantifier-free formula.

A formal proof could be given using induction, but the preceding example illustrates the general method.

Recall that a universal ($\forall_1$) formula is a first-order prenex formula all of whose quantifiers are universal: $\forall x_1 \forall x_2 \cdots \forall x_k \, \alpha$, where α is quantifier-free. Similarly, an existential ($\exists_1$) formula is a first-order prenex formula all of whose quantifiers are existential.

COROLLARY 42A   For any first-order φ, we can find a universal formula θ in an expanded language containing function symbols, such that φ is satisfiable iff θ is satisfiable.

By applying this corollary to $\neg\varphi$, we obtain an existential formula (with function symbols) that is valid iff φ is valid.

PROOF.   Again we will only illustrate the situation by an example. Say that φ is

$$\exists y_1 \forall x_1 \exists y_2 \forall x_2 \forall x_3 \exists y_3 \, \psi(y_1, y_2, y_3).$$

First we replace φ by the logically equivalent formula in Skolem form:

$$\exists y_1 \exists F_2 \exists F_3 \forall x_1 \forall x_2 \forall x_3 \, \psi(y_1, F_2 x_1, F_3 x_1 x_2 x_3).$$

Then for θ we take

$$\forall x_1 \forall x_2 \forall x_3 \, \psi(c, f x_1, g x_1 x_2 x_3),$$

where c, f, and g are new function symbols having zero, one, and three places, respectively. In general θ is *not* logically equivalent to φ. But we do have $\theta \models \varphi$ (in the expanded language). And any model $\mathfrak{A}$ of φ can be expanded (by defining $c^{\mathfrak{A}}$, $f^{\mathfrak{A}}$, and $g^{\mathfrak{A}}$ correctly) to be a model of θ. Thus φ and θ are "equally satisfiable."   ⊣

This result reduces the general problem of testing first-order formulas for satisfiability to the special case of universal formulas (with function symbols). And by the same token, it reduces the problem of testing for validity to the $\exists_1$ case. From these reductions we can derive an undecidability result for first-order logic:

COROLLARY 42B   Consider a recursively numbered language having a two-place predicate symbol and infinitely many k-place function symbols for each $k \geq 0$.

(a) The set of Gödel numbers of satisfiable universal (first-order) sentences is not recursive.

(b) The set of Gödel numbers of valid existential (first-order) sentences is not recursive.

PROOF.   (b) Given any sentence σ we can, by applying Corollary 42A to $\neg\sigma$, effectively find an existential sentence that is valid iff σ is valid. Hence a decision procedure for the existential validities would yield a decision procedure for arbitrary validities, in contradiction to Church's theorem.   ⊣

We can use predicate variables instead of function variables in these results, but at a price. Suppose we begin with a first-order formula. It is equivalent to a formula ψ in Skolem normal form; suppose for simplicity that $\psi = \exists F\,\varphi$, where φ has only individual quantifiers and F is a one-place function variable. We can choose φ in such a way that F occurs only in equations of the form $u = Ft$ (for terms t and u not containing F). This can be done by replacing, for example, an atomic formula $\alpha(Ft)$ by either $\forall x(x = Ft \to \alpha(x))$ or $\exists x(x = Ft \wedge \alpha(x))$.

Next observe that a formula

$$\exists F\_u = Ft\_,$$

wherein F occurs only in the form shown, is equivalent to

$$\exists X(\forall y\,\exists!z\,Xyz \wedge \_Xtu\_).$$

If one pursues this question (as we will not do here) one finds that any first-order formula is logically equivalent to a second-order formula consisting of

(a) A string of existential predicate quantifiers, followed by
(b) A string of universal individual quantifiers, followed by
(c) A string of existential individual quantifiers, followed by
(d) A quantifier-free formula.

There are corresponding versions — see Exercise 4 — of Corollaries 42A and 42B. The analogue of Corollary 42A reduces the problem of testing first-order formulas for satisfiability to the special case of $\forall_2$ formulas (with predicate symbols). The problem of testing for validity is reduced to the $\exists_2$ case.

The analogue of Corollary 42B can be compared with Exercise 10 in Section 2.6, where it is shown that the set of $\forall_2$ validities without function symbols is decidable.

Herbrand Expansions

We have seen (in Corollary 42A) how, given a formula of first-order logic, to find an "equally satisfiable" universal formula. And thus

(Corollary 42B) the question of satisfiability in first-order logic is
reducible to the question of satisfiability of universal formulas.

Now we go one step further: satisfiability of these universal formulas
is reducible — but in a weaker sense — to satisfiability in *sentential*
logic.

> **EXAMPLE.**   We know that $\forall x \exists y\, Pxy \not\models \exists y \forall x\, Pxy$. But pretend
> we did not know this, and that we are interested in determin-
> ing whether or not logic implication holds here. That is equiv-
> alent to determining whether or not the hypothesis $\forall x \exists y Pxy$
> together with the negation of the conclusion $\neg \exists y \forall x\, Pxy$ is
> unsatisfiable.
>
> By the Skolem normal form theorem, we can replace these sen-
> tences by certain logically equivalent sentences; we wish to deter-
> mine whether or not $\exists F \forall x\, PxFx$ together with $\exists G \forall y \neg PGyy$
> is unsatisfiable. And as in Corollary 42A, we replace these sen-
> tences by equally satisfiable universal sentences; we wish to de-
> termine whether or not the set $\{\forall x\, Pxfx,\ \forall y\, Pgy\, y\}$ is unsat-
> isfiable (where f and g are new function symbols).
>
> But this set of universal sentences *is* satisfiable, and moreover,
> the set can be made to generate its own model. Here is how. For
> the universe of our model we will use the *Herbrand universe H*,
> which is the set of all terms (in the language with f and g). Thus
> H contains, for each variable u, the terms
>
> $$u,\ fu,\ gu,\ ffu,\ fgu, \ldots.$$
>
> Let Δ be the set of all *instances* of the universal sentences, that
> is, the formulas obtained by dropping all the universal quantifiers
> and plugging in (for the universally quantified variables) arbitrary
> terms from the Herbrand universe. Thus Δ contains, for each
> variable u, the quantifier-free formulas
>
> $$Pufu,\ Pgufgu, \ldots, \ \neg Pguu, \ \neg Pgfufu, \ldots.$$
>
> Now we consider Δ from the point of view of *sentential* logic.
> The sentence symbols are the atomic formulas, e.g., $Pgfufu$.
> And in this example Δ is satisfiable in sentential logic. That is,
> there is a truth assignment v on the set of sentence symbols so
> that $\overline{v}(\alpha) = T$ for every α in Δ. Here is one such v:
>
> $$v(Pt_1t_2) = \begin{cases} T & \text{if } t_1 \text{ is shorter than } t_2 \\ F & \text{if } t_1 \text{ is at least as long as } t_2 \end{cases}$$
>
> Finally, we use this truth assignment v (in sentential logic) to
> make a structure $\mathfrak{H}$ (in first-order logic) that will be a model of
> the universal sentences. The universe is the Herbrand universe:

$|\mathfrak{H}| = H$. (There are echos of the completeness proof of Section 2.5 here.) The function symbols are interpreted autonymously — as naming themselves: $f^{\mathfrak{H}}(t)$ is ft and $g^{\mathfrak{H}}(t)$ is gt. Where v comes in is to interpret the predicate symbol P:

$$\langle t_1, t_2 \rangle \in P^{\mathfrak{H}} \iff v(Pt_1 t_2) = T$$

This structure works. First $\models_{\mathfrak{H}} \forall x \, Pxfx$ because for every term t in the Herbrand universe, $\langle t, ft \rangle \in P^{\mathfrak{H}}$. Secondly $\models_{\mathfrak{H}} \forall y \neg Pgy \, y$ because for every term t in the Herbrand universe, $\langle gt, t \rangle \notin P^{\mathfrak{H}}$.

We conclude that the hypothesis $\forall x \exists y \, Pxy$ together with the negated conclusion $\neg \exists y \forall x \, Pxy$ is indeed satisfiable, and hence $\forall x \exists y \, Pxy \not\models \exists y \forall x \, Pxy$.

To what extent can we generalize from this example? Assume, for simplicity, that the language does not contain equality. (Exercise 7 indicates the changes needed to accommodate equality.) Suppose we want to determine whether or not $\Gamma \models \varphi$ for a set Γ; φ of formulas in first-order logic. This is equivalent to determining whether or not the set Γ; $\neg \varphi$ is unsatisfiable.

We can replace each of the formulas here by a logically equivalent formula in Skolem normal form. And as in Corollary 42A, we get an equally satisfiable set Ψ of universal formulas. (In applying Skolem normal form, we use *different* Skolem function symbols for each formula, so that there is no clash between formulas.) This brings us to the situation:

$$\Gamma \models \varphi \iff \Psi \text{ is unsatisfiable}$$

and Ψ is a set of universal formulas.

Let H be the Herbrand universe, that is, the set of all terms in the language of Ψ. Let Δ be the set of all instances of the universal formulas in Ψ (i.e., formulas obtained by dropping all the universal quantifiers and plugging in for the universally quantified variables arbitrary terms from the Herbrand universe). Then Δ consists of quantifier-free formulas only. We consider Δ from the viewpoint of sentential logic, where the sentence symbols are the atomic formulas.

Case I: Δ is unsatisfiable in sentential logic. In this case, we can conclude that Ψ is unsatisfiable and $\Gamma \models \varphi$ in first-order logic. This is because a universal formula logically implies all of its instances. Therefore $\Psi \models \delta$ for every δ in Δ (in first-order logic). Any model of Ψ must be a model of Δ. But from a model $\mathfrak{A}$ of Δ we can extract a truth assignment v that satisfies Δ in sentential logic. (Remember Exercise 3 in Section 2.4. Note the interesting interplay between first-order logic and sentential logic.)

Case II: Δ is satisfiable in sentential logic, say by the truth assignment v. Then we will use v to make a structure $\mathfrak{H}$ in which Ψ is satisfiable and $\Gamma \not\models \varphi$ because $\mathfrak{H}$ will give a counterexample.

As in the example, the universe $|\mathfrak{H}|$ is the Herbrand universe H, the set of all terms in the language of Ψ. And again the function symbols are interpreted autonymously: $f^{\mathfrak{H}}(t_1, \ldots, t_n) = f t_1 \cdots t_n$. To interpret a predicate symbol P we use the truth assignment v:

$$\langle t_1, \ldots, t_n \rangle \in P^{\mathfrak{H}} \iff v(P t_1 \cdots t_n) = T$$

Then we claim that every formula in Ψ is satisfied in $\mathfrak{H}$ by the identity function $s(x) = x$ on the variables. First, note that $\bar{s}(t) = t$ for any term t in H; we had the same situation in step 4 of the completeness proof in Section 2.5. Secondly, note that for an atomic formula $P t_1 \cdots t_n$,

$$\models_{\mathfrak{H}} P t_1 \cdots t_n[s] \iff \langle t_1, \ldots, t_n \rangle \in P^{\mathfrak{H}} \iff v(P t_1 \cdots t_n) = T.$$

By Exercise 3 of Section 2.4 again, any formula δ in Δ is satisfied in $\mathfrak{H}$ with s (because $\bar{v}(\delta) = T$).

Consider any formula in Ψ. It is a universal formula; to simplify the notation, say it is $\forall v_1 \forall v_2 \, \theta(v_1, v_2, v_3)$, where θ is quantifier-free. We need to check for any terms t_1 and t_2 in H that $\models_{\mathfrak{H}} \theta[\![t_1, t_2, v_3]\!]$. This is equivalent (by the substitution lemma) to saying that the formula $\theta(t_1, t_2, v_3)$ is satisfied in $\mathfrak{H}$ by s. But this formula is an instance of $\forall v_1 \forall v_2 \, \theta(v_1, v_2, v_3)$, and so $\theta(t_1, t_2, v_3)$ is in Δ. As noted above, our construction was arranged so that every formula in Δ is satisfied in $\mathfrak{H}$ with s. This is what we needed.

We can summarize the result as follows. For simplicity, the result is stated only for sentences.

> **HERBRAND'S THEOREM**   Consider a set Γ; φ of sentences in a first-order language without equality. Let Δ be as above. Then either (case I) Δ is unsatisfiable in sentential logic and $\Gamma \models \varphi$, or (case II) Δ is satisfiable in sentential logic and the structure $\mathfrak{H}$ constructed above is a model of Γ in which φ is false.

(Herbrand's work was in his 1930 dissertation, completed not long before he was killed in a mountain-climbing accident. His statement of the theorem was quite different from this one, but the ideas follow his work, and 1928 work of Thoralf Skolem.)

In case I, by the compactness theorem of sentential logic, some finite subset of Δ is unsatisfiable. This fact can be used to obtain a proof of the compactness theorem for first-order logic, not relying on Section 2.5 or the deductive calculus of Section 2.4.

Moreover, a proof of the enumerability theorem, similarly independent of Sections 2.4 and 2.5, can be extracted from the Herbrand approach. Take the special case $\Gamma = \varnothing$. If φ is valid then as we produce

more and more of Δ, at some point we will have an unsatisfiable set, a fact we can recognize by use of truth-tables. If φ is not valid, then as we produce more and more of Δ, we are making a structure in which φ fails, but the structure is infinite and the construction never terminates.

Exercises

1. Prove the Löwenheim–Skolem theorem in the following improved form: Let $\mathfrak{A}$ be a structure for a countable language. Let S be a countable subset of $|\mathfrak{A}|$. Then there is a countable *substructure* $\mathfrak{B}$ of $\mathfrak{A}$ with $S \subseteq |\mathfrak{B}|$ with the property that for any function s mapping the variables into $|\mathfrak{B}|$ and any (first-order) φ,

$$\models_\mathfrak{A} \varphi[s] \quad \text{iff} \quad \models_\mathfrak{B} \varphi[s].$$

   Suggestion: Choose Skolem functions for all formulas. Close S under the functions. *Remark*: A substructure $\mathfrak{B}$ with this property is said to be an *elementary* substructure. Note that the property implies (by taking φ to be a sentence) that $\mathfrak{A} \equiv \mathfrak{B}$. On the one hand, this form gives a stronger conclusion than we had in Section 2.6. Not only do we get that Th $\mathfrak{A}$ has *some* countable model, we get a countable *sub*model. On the other hand, the proof uses the axiom of choice.

2. Extend the previous exercise to the uncountable case. Assume that $\mathfrak{A}$ is a structure for a language of cardinality λ. Let S be a subset of $|\mathfrak{A}|$ having cardinality κ. Show that there is an elementary substructure $\mathfrak{B}$ of $\mathfrak{A}$ of cardinality at most $\kappa + \lambda$ with $S \subseteq |\mathfrak{B}|$.

3. Show that Corollary 42B is optimal in the following sense:
   (a) Given any $\exists_1$ sentence σ we can effectively decide whether or not σ is satisfiable.
   (b) Given any $\forall_1$ sentence σ we can effectively decide whether or not σ is valid.

4. (a) State the two corollaries (analogous to 42A and 42B) described at the end of this section.
   (b) Supply proofs.

5. Repeat the example given for Herbrand expansions, but for the converse: $\exists y \forall x \, Pxy \models \forall x \exists y \, Pxy$. Show that in this case the set Δ is unsatisfiable in sentential logic.

6. Apply the method of Herbrand expansions to establish the following: $\models \exists x (Px \rightarrow \forall x \, Px)$.

7. Modify the Herbrand expansion construction to accomodate a language with equality. *Suggestion*: In effect, step 5 of the completeness proof in Section 2.5 must be added. Add enough universal sentences to assure that $\{\langle t_1, t_2 \rangle | \, v(t_1 = t_2) = T$ is a congruence relation.

SECTION 4.3
Many-Sorted Logic

We now return to first-order languages, but with many sorts of variables, ranging over different universes. (In the next section this will be applied to the case in which one sort of variable is for elements of a universe, another for subsets of that universe, yet another for binary relations, and so forth.)

In informal mathematics one sometimes says things like, "We use Greek letters for ordinals, capital script letters for sets of integers, ..." In effect, one thereby adopts several sorts of variables, each sort having its own universe. We now undertake to examine this situation precisely. As might be expected, nothing is drastically different from the usual one-sorted situation. None of the results of this section are at all deep, and most of the proofs are omitted.

Assume that we have a nonempty set I, whose members are called sorts, and symbols arranged as follows:

- A. *Logical symbols*
    - 0. Parentheses: (,).
    - 1. Sentential connective symbols: $\neg$, $\rightarrow$.
    - 2. Variables: For each sort i, there are variables v_1^i, v_2^i, ... of sort i.
    - 3. Equality symbols: For some $i \in I$ there may be the symbol $=_i$, said to be a predicate symbol of sort $\langle i, i \rangle$.
- B. *Parameters*
    - 0. Quantifier symbols: For each sort i there is a universal quantifier symbol $\forall_i$.
    - 1. Predicate symbols: For each $n > 0$ and each n-tuple $\langle i_1, \ldots, i_n \rangle$ of sorts, there is a set (possibly empty) of n-place predicate symbols, each of which is said to be of sort $\langle i_1, \ldots, i_n \rangle$.
    - 2. Constant symbols: For each sort i there is a set (possibly empty) of constant symbols each of which is said to be of sort i.
    - 3. Function symbols: For each $n > 0$ and each $(n + 1)$-tuple $\langle i_1, \ldots, i_n, i_{n+1} \rangle$ of sorts, there is a set (possibly empty) of n-place function symbols, each of which is said to be of sort $\langle i_1, \ldots, i_n, i_{n+1} \rangle$.

As usual, we must assume that these categories of symbols are disjoint, and further that no symbol is a finite sequence of other symbols.

Each term will be assigned a unique sort. We define the set of terms of sort i inductively, simultaneously for all i:

1. Any variable of sort i or constant symbol of sort i is a term of sort i.

2. If $t_1, \ldots, t_n$ are terms of sort $i_1, \ldots, i_n$, respectively, and f is a function symbol of sort $\langle i_1, \ldots, i_n, i_{n+1} \rangle$ then $f t_1 \cdots t_n$ is a term of sort i_{n+1}.

This definition can be recast into a more familiar form. The set of pairs $\langle t, i \rangle$ such that t is a term of sort i is built up from (i.e., generated from) the basic set

$$\{ \langle v_n^i, i \rangle \mid n \geq 1 \ \& \ i \in I \} \cup \{ \langle c, i \rangle \mid c \text{ is a constant symbol of sort } i \}$$

by the operations that, for a function symbol f of sort $\langle i_1, \ldots, i_n, i_{n+1} \rangle$, produce the pair $\langle f t_1 \cdots t_n, i_{n+1} \rangle$ from the pairs $\langle t_1, i_1 \rangle, \ldots, \langle t_n, i_n \rangle$.

An atomic formula is a sequence $P t_1 \cdots t_n$ consisting of a predicate symbol of sort $\langle i_1, \ldots, i_n \rangle$ and terms $t_1, \ldots, t_n$ of sort $i_1, \ldots, i_n$, respectively. The nonatomic formulas are then formed using the connectives $\neg$, $\rightarrow$ and the quantifiers $\forall_i \, v_n^i$.

A many-sorted structure $\mathfrak{A}$ is a function on the set of parameters which assigns to each the correct type of object:

1. To the quantifier symbol $\forall_i$, $\mathfrak{A}$ assigns a nonempty set $|\mathfrak{A}|_i$ called the *universe* of $\mathfrak{A}$ of sort i.

2. To each predicate symbol P of sort $\langle i_1, \ldots, i_n \rangle$, $\mathfrak{A}$ assigns a relation

$$P^{\mathfrak{A}} \subseteq |\mathfrak{A}|_{i_1} \times \cdots \times |\mathfrak{A}|_{i_n}.$$

3. To each constant symbol c of sort i, $\mathfrak{A}$ assigns a point $c^{\mathfrak{A}}$ in $|\mathfrak{A}|_i$.

4. To each function symbol f of sort $\langle i_1, \ldots, i_n, i_{n+1} \rangle$, $\mathfrak{A}$ assigns a function

$$f^{\mathfrak{A}} : |\mathfrak{A}|_{i_1} \times \cdots \times |\mathfrak{A}|_{i_n} \rightarrow |\mathfrak{A}|_{i_{n+1}}.$$

The definitions of truth and satisfaction are the obvious ones, given that $\forall_i$ is to mean "for all members of the universe $|\mathfrak{A}|_i$ of sort i."

In a many-sorted structure, the universes of the various sorts might or might not be disjoint. But since we have no equality symbols *between* sorts, any nondisjointness must be regarded as accidental. In particular, there will always be an elementarily equivalent structure whose universes are disjoint.

Reduction to One-Sorted Logic

Many-sorted languages may at times be convenient (as in the following section). But there is nothing essential that can be done with them that cannot already be done without them. We now proceed to make this assertion in a more precise form.

We will consider a one-sorted language having all the predicate, constant, and function symbols of our assumed many-sorted language. In addition, it will have a one-place predicate symbol Q_i for each i in I. There is a syntactical translation taking each many-sorted formula φ into a one-sorted formula φ^*. In this translation all equality symbols are replaced by $=$. The only other change is in the quantifiers (the quantifier symbols and the quantified variables): We replace

$$\forall_i v_n^i \_\_ v_n^i \_\_$$

by

$$\forall v(Q_i v \rightarrow \_\_v\_\_),$$

where v is a variable chosen not to conflict with the others. Thus the quantifiers of sort i are "relativized" to Q_i. (The free variables are left alone.)

Turning now to semantics, we can convert a many-sorted structure $\mathfrak{A}$ into a structure $\mathfrak{A}^*$ for the above one-sorted language. The universe $|\mathfrak{A}^*|$ is the union $\bigcup_{i \in I} |\mathfrak{A}|_i$ of all the universes of $\mathfrak{A}$. To Q_i is assigned the set $|\mathfrak{A}|_i$. On the predicate and constant symbols, $\mathfrak{A}^*$ agrees with $\mathfrak{A}$. For a function symbol f, the function $f^{\mathfrak{A}^*}$ is an arbitrary extension of $f^{\mathfrak{A}}$. (Of course this last sentence does not completely specify $f^{\mathfrak{A}^*}$. The results we give for $\mathfrak{A}^*$ hold for any structure obtained in the manner just described.)

> **LEMMA 43A**   A many-sorted sentence σ is true in $\mathfrak{A}$ iff σ^* is true in $\mathfrak{A}^*$.

To prove this, one makes a stronger statement concerning formulas:

$$\models_{\mathfrak{A}} \varphi[s] \iff \models_{\mathfrak{A}^*} \varphi^*[s]$$

where $s(v_n^i) \in |\mathfrak{A}|_i$. The stronger statement is then proved by induction.

Consider now the other direction. A one-sorted structure is not always convertible into a many-sorted structure. So we will impose some conditions. Let Φ be the set consisting of the following one-sorted sentences:

1. $\exists v Q_i v$, for each i in I.
2. $\forall v_1 \cdots \forall v_n (Q_{i_1} v_1 \rightarrow \cdots \rightarrow Q_{i_n} v_n \rightarrow Q_{i_{n+1}} f v_1 \cdots v_n)$, for each function symbol f of sort $\langle i_1, \ldots, i_n, i_{n+1} \rangle$. We include the case $n = 0$, in which case the above becomes the sentence $Q_i c$ for a constant symbol c of sort i.

Notice that the above $\mathfrak{A}^*$ was a model of Φ. A one-sorted model $\mathfrak{B}$ of Φ does convert into a many-sorted $\mathfrak{B}^\sharp$. The conversion is performed

in the natural way:

$$|\mathfrak{B}^{\sharp}|_i = Q_i^{\mathfrak{B}};$$
$$P^{\mathfrak{B}^{\sharp}} = P^{\mathfrak{B}} \cap (Q_{i_1}^{\mathfrak{B}} \times \cdots \times Q_{i_n}^{\mathfrak{B}}), \text{ where } P \text{ is a predicate symbol}$$
$$\quad \text{of sort } \langle i_1, \ldots, i_n \rangle$$
$$c^{\mathfrak{B}^{\sharp}} = c^{\mathfrak{B}};$$
$$f^{\mathfrak{B}^{\sharp}} = f^{\mathfrak{B}} \cap (Q_{i_1}^{\mathfrak{B}} \times \cdots \times Q_{i_n}^{\mathfrak{B}} \times Q_{i_{n+1}}^{\mathfrak{B}}), \text{ the restriction of}$$
$$\quad f^{\mathfrak{B}} \text{ to } Q_{i_1}^{\mathfrak{B}} \times \cdots \times Q_{i_n}^{\mathfrak{B}}, \text{ where } f \text{ is a function symbol}$$
$$\quad \text{of sort } \langle i_1, \ldots, i_n, i_{n+1} \rangle.$$

LEMMA 43B   If $\mathfrak{B}$ is a model of Φ, then $\mathfrak{B}^{\sharp}$ is a many-sorted structure. Furthermore, a many-sorted sentence σ is true in $\mathfrak{B}^{\sharp}$ iff σ^* is true in $\mathfrak{B}$.

The proof is similar to the proof of Lemma 43A.

Notice that $\mathfrak{B}^{\sharp*}$ is not in general equal to $\mathfrak{B}$. (For example, $|\mathfrak{B}|$ may contain points not belonging to any $Q_i^{\mathfrak{B}}$.) On the other hand, $\mathfrak{A}^{*\sharp}$ is equal to $\mathfrak{A}$.

THEOREM 43C   In the many-sorted language

$$\Sigma \models \sigma$$

iff in the one-sorted language

$$\Sigma^* \cup \Phi \models \sigma^*.$$

PROOF.   ($\Rightarrow$) Assume that $\Sigma \models \sigma$ and let $\mathfrak{B}$ be a one-sorted model of $\Sigma^* \cup \Phi$ (where $\Sigma^* = \{\sigma^* \mid \sigma \in \Sigma\}$). Then $\mathfrak{B}^{\sharp}$ is a model of Σ by Lemma 43B. Hence $\mathfrak{B}^{\sharp}$ is a model of σ. So by Lemma 43B again, $\mathfrak{B}$ is a model of σ^*.

($\Leftarrow$)   Similar, with Lemma 43A.   ⊣

By using Theorem 43C, we can now infer the following three theorems from the corresponding one-sorted results.

COMPACTNESS THEOREM   If every finite subset of a set Σ of many-sorted sentences has a model, then Σ has a model.

PROOF.   Assume that every finite subset Σ_0 of Σ has a many-sorted model $\mathfrak{A}_0$. Then a finite subset Σ_0^* of Σ^* has the model $\mathfrak{A}_0^*$. Hence by the ordinary compactness theorem, Σ^* has a model $\mathfrak{B}$. $\mathfrak{B}^{\sharp}$ is then a model of Σ.   ⊣

ENUMERABILITY THEOREM   For a recursively numbered many-sorted language, the set of Gödel numbers of valid sentences is recursively enumerable.

PROOF.   For a many-sorted σ, we have by Theorem 43C,

$$\models \sigma \quad \text{iff} \quad \Phi \models \sigma^*.$$

Since Φ is recursive, $\mathrm{Cn}\,\Phi$ is recursively enumerable. And σ^* depends recursively on σ, so we can apply Exercise 7(b) of Section 3.5.   ⊣

LÖWENHEIM–SKOLEM THEOREM   For any many-sorted structure (for a countable language) there is an elementarily equivalent countable structure.

PROOF.   Say that the given structure is $\mathfrak{A}$. Then $\mathfrak{A}^*$ is a one-sorted model of $(\mathrm{Th}\,\mathfrak{A})^* \cup \Phi$. Hence by the ordinary Löwenheim–Skolem theorem, $(\mathrm{Th}\,\mathfrak{A})^* \cup \Phi$ has a countable model $\mathfrak{B}$. $\mathfrak{B}^\sharp$ is a model of $\mathrm{Th}\,\mathfrak{A}$ and so is elementarily equivalent to $\mathfrak{A}$.   ⊣

SECTION 4.4
General Structures

We now return to the discussion of second-order logic begun in Section 4.1. We discussed there (a) the syntax, i.e., the set of wffs for second order, and (b) the semantics, i.e., the concept of structure (which was the same as for first order) and the definition of satisfaction and truth.

In this section we want to leave (a) unchanged, but we want to present an alternative to (b). The idea can be stated very briefly: We view the language (previously thought of as second-order) now as being a many-sorted elementary (i.e., first-order) language. The result is to make open to interpretation not only the universe over which individual variables range but also the universes for the predicate and function variables. This approach is particularly suited to number theory; that case is examined briefly at the end of this section.

The Many-Sorted Language

Despite the fact that we want ultimately to consider the grammar of Section 4.1, it will be expedient to consider also a many-sorted (first-order) language constructed from the second-order language of Section 4.1. We take $\aleph_0$ sorts: the one individual sort (with variables $v_1, v_2, \ldots$); for each $n > 0$, the n-place predicate sort (with variables $X_1^n, X_2^n, \ldots$); and for each $n > 0$, the n-place function sort (with variables $F_1^n, F_2^n, \ldots$). We will use equality ($=$) only between terms of the individual sort. The predicate and function parameters of our assumed second-order language will also be parameters of the many-sorted language, and will take as arguments terms of the individual sort. (For a function parameter f, the term $f\vec{t}$ is of the individual sort. The only terms of predicate or function sort are the variables of those sorts.)

In addition, we now use two new classes of parameters. For each $n > 0$ there is a *membership* predicate parameter ε_n which takes as

arguments one term of the n-place predicate sort (i.e., a variable X_m^n) and n terms of the individual sort. Thus, for example,

$$\varepsilon_3 X^3 v_2 v_1 v_8$$

is a wff. Its intended interpretation is that the triple denoted by $\langle v_2, v_1, v_8 \rangle$ is to belong to the relation denoted by X^3. This is exactly the interpretation assigned previously to the second-order formula

$$X^3 v_2 v_1 v_8,$$

and, in fact, readers are advised to identify these two formulas closely in their minds.

For each $n > 0$, there is also the *evaluation* function parameter E_n. E_n takes as arguments one term of the n-place function sort (i.e., a variable F_m^n) and n terms of the individual sort. The resulting term,

$$E_n F^n t_1 \cdots t_n,$$

is itself of the individual sort. Again readers are advised to identify closely the term $E_n F^n t_1 \cdots t_n$ with the previous $F^n t_1 \cdots t_n$.

There is an obvious way of translating between the second-order language of Section 4.1 and the present many-sorted language. In one direction we stick on the ε_n and E_n symbols; in the other direction we take them off. The purpose of these symbols is to make the language conform to Section 4.3.

A many-sorted structure has universes for each sort and assigns suitable objects to the various parameters (as described in the preceding section). First, we want to show that without loss of generality, we may suppose that ε_n is interpreted as genuine membership and E_n as genuine evaluation.

THEOREM 44A   Let $\mathfrak{A}$ be a structure for the above many-sorted language such that the different universes of $\mathfrak{A}$ are disjoint. Then there is a homomorphism h of $\mathfrak{A}$ onto a structure $\mathfrak{B}$ such that

(a) h is one-to-one, in fact the identity, on the individual universe (from which it follows that

$$\models_{\mathfrak{A}} \varphi[s] \quad \text{iff} \quad \models_{\mathfrak{B}} \varphi[h \circ s]$$

for each formula φ).

(b) The n-place predicate universe of $\mathfrak{B}$ consists of certain n-ary relations over the individual universe, and $\langle R, a_1, \ldots, a_n \rangle$ is in $\varepsilon_n^{\mathfrak{B}}$ iff $\langle a_1, \ldots, a_n \rangle \in R$.

(c) The n-place function universe of $\mathfrak{B}$ consists of certain n-place functions on the individual universe, and $E_n^{\mathfrak{B}}(f, a_1, \ldots, a_n) = f(a_1, \ldots, a_n)$.

PROOF.   Since the universes of $\mathfrak{A}$ are disjoint, we can define h on one universe at a time. On the individual universe U, h is the identity.

On the universe of the n-place predicate sort,

$$h(Q) = \{\langle a_1, \ldots, a_n \rangle \mid \text{each } a_i \text{ is in } U \text{ and}$$
$$\langle Q, a_1, \ldots, a_n \rangle \text{ is in } \varepsilon_n^{\mathfrak{A}}\}.$$

Thus

$$\langle a_1, \ldots, a_n \rangle \in h(Q) \quad \text{iff} \quad \langle Q, a_1, \ldots, a_n \rangle \text{ is in } \varepsilon_n^{\mathfrak{A}}. \quad (1)$$

Similarly, on the universe of the n-place function sort,

$$h(g) \text{ is the } n\text{-place function on } U \text{ whose value at}$$
$$\langle a_1, \ldots, a_n \rangle \text{ is } E_n^{\mathfrak{A}}(g, a_1, \ldots, a_n).$$

Thus

$$h(g)(a_1, \ldots, a_n) = E_n^{\mathfrak{A}}(g, a_1, \ldots, a_n). \quad (2)$$

For $\varepsilon_n^{\mathfrak{B}}$ we take simply the membership relation,

$$\langle R, a_1, \ldots, a_n \rangle \text{ is in } \varepsilon_n^{\mathfrak{B}} \quad \text{iff} \quad \langle a_1, \ldots, a_n \rangle \in R. \quad (3)$$

For $E_n^{\mathfrak{B}}$ we take the evaluation function,

$$E_n^{\mathfrak{B}}(f, a_1, \ldots, a_n) = f(a_1, \ldots, a_n). \quad (4)$$

On the other parameters (inherited from the second-order language) $\mathfrak{B}$ agrees with $\mathfrak{A}$.

Then it is clear upon reflection that h is a homomorphism of $\mathfrak{A}$ onto $\mathfrak{B}$. That h preserves ε_n follows from (1) and (3), where in (3) we take $R = h(Q)$. Similarly, from (2) and (4) it follows that h preserves E_n.

Finally, we have to verify the parenthetical remark of part (a). This follows from the many-sorted analogue of the homomorphism theorem of Section 2.2, by using the fact that we have equality only for the individual sort, where h is one-to-one.   ⊣

By the above theorem, we can restrict attention to structures $\mathfrak{B}$ in which ε_n and E_n are fixed by (b) and (c) of the theorem. But since $\varepsilon_n^{\mathfrak{B}}$ and $E_n^{\mathfrak{B}}$ are determined by the rest of $\mathfrak{B}$, we really do not need them at all. When we discard them, we have a general pre-structure for our second-order grammar.

General Structures for Second-Order Languages

These structures provide the alternative semantics mentioned at the beginning of this section.

> DEFINITION.   A *general pre-structure* $\mathfrak{A}$ for our second-order language consists of a structure (in the original sense), together with the additional sets:
>
>   (a) For each $n > 0$, an n-place *relation universe*, which is a set of n-ary relations on $|\mathfrak{A}|$;

(b) For each $n > 0$, an n-place *function universe*, which is a set of functions from $|\mathfrak{A}|^n$ into $|\mathfrak{A}|$.

$\mathfrak{A}$ is a *general structure* if, in addition, all comprehension sentences are true in $\mathfrak{A}$.

The last sentence of the definition requires explanation. First, a comprehension sentence is a sentence obtained as a generalization of a comprehension formula (see Example 3 of Section 4.1). Thus it is a generalization of

$$\exists X^n \, \forall v_1 \cdots \forall v_n (X^n v_1 \cdots v_n \leftrightarrow \varphi),$$

where X^n does not occur free in φ, or a generalization of

$$\forall v_1 \cdots \forall v_n \, \exists! v_{n+1} \, \psi \rightarrow$$
$$\exists F^n \, \forall v_1 \cdots \forall v_{n+1}(F^n v_1 \cdots v_n = v_{n+1} \leftrightarrow \psi),$$

where F^n does not occur free in ψ. (Here φ and ψ can have individual variables, predicate variables, and function variables.)

Next we must say what it means for a comprehension sentence (or any second-order sentence for that matter) to be true in $\mathfrak{A}$. Assume then that $\mathfrak{A}$ is a general pre-structure. Then a sentence σ is true in $\mathfrak{A}$ iff the result of converting σ into a many-sorted sentence (by adding ε_n and E_n) is true in $\mathfrak{A}$, with ε_n interpreted as membership and E_n as evaluation.

More generally, let φ be a second-order formula, and let s be a function that assigns to each individual variable a member of $|\mathfrak{A}|$, to each predicate variable a member of the relation universe of $\mathfrak{A}$, and to each function variable a member of the function universe of $\mathfrak{A}$. Then we say that $\mathfrak{A}$ satisfies φ with s (written $\models^G_\mathfrak{A} \varphi[s]$) iff the many-sorted version of φ is satisfied with s in the structure $\mathfrak{A}$, where ε_n is interpreted as membership and E_n as evaluation.

The essential consequences of this definition of satisfaction are the following, which should be compared with 5 and 6 of page 283.

$\models^G_\mathfrak{A} \forall X^n \, \varphi[s]$ iff for every R in the n-place relation universe of $\mathfrak{A}$, $\models^G_\mathfrak{A} \varphi[s(X^n \mid R)]$.

$\models^G_\mathfrak{A} \forall F^n \, \varphi[s]$ iff for every f in the n-place relation universe of $\mathfrak{A}$, $\models^G_\mathfrak{A} \varphi[s(F^n \mid f)]$.

This, then, is the alternative approach mentioned at the beginning of the section. It involves treating the second-order grammar as being a many-sorted first-order grammar in disguise. Because this approach is basically first-order, we have the Löwenheim–Skolem theorem, the compactness theorem, and the enumerability theorem.

LÖWENHEIM–SKOLEM THEOREM   If the set Σ of sentences in a countable second-order language has a general model, then it has a countable general model.

Here a countable general model is one in which every universe is countable (or equivalently, the union of all the universes is countable).

PROOF.   Let Γ be the set of comprehension sentences. Then $\Sigma \cup \Gamma$, viewed as a set of many-sorted sentences, has a countable many-sorted model by the Löwenheim–Skolem theorem of the preceding section. By Theorem 44A, a homomorphic image of that model is a general pre-structure satisfying $\Sigma \cup \Gamma$, and hence is a general model of Σ.   ⊣

COMPACTNESS THEOREM   If every finite subset of a set Σ of second-order sentences has a general model, then Σ has a general model.

PROOF.   The proof is exactly as above. Every finite subset of $\Sigma \cup \Gamma$ has a many-sorted model, so we can apply the compactness theorem of the preceding section.   ⊣

ENUMERABILITY THEOREM   Assume that the language is recursively numbered. Then the set of Gödel numbers of second-order sentences that are true in every general structure is recursively enumerable.

PROOF.   A sentence σ is true in every general structure iff it is a many-sorted consequence of Γ. And $\sharp\Gamma$ is recursive.   ⊣

The above two theorems assure us that there is an acceptable deductive calculus such that τ is deducible from Σ iff τ is true in every general model of Σ (see the remarks at the beginning of Section 2.4). But now that we know there is such a complete deductive calculus, there is no compelling reason to go into the detailed development of one.

We can compare the two approaches to second-order semantics as follows: The version of Section 4.1 (which we will now call *absolute* second-order logic) is a hybrid creature, in which the meaning of the parameters is left open to interpretation by structures, but the concept of being (for example) a subset is not left open, but is treated as having a fixed meaning. The version of the present section (*general* second-order logic) avoids appealing to a fixed notion of subset, and consequently is reducible to first-order logic. In this respect it is like axiomatic set theory, where one speaks of sets and sets of sets and so forth, but the theory is a first-order theory.

By enlarging the class of structures, general second-order logic diminishes the cases in which logical implication holds. That is, if every general model of Σ is a general model of σ, it then follows that $\Sigma \models \sigma$ in absolute second-order logic. But the converse fails. For example, take $\Sigma = \varnothing$: The set of sentences true in all general models is a recursively enumerable subset of the nonarithmetical set of valid sentences of absolute second-order logic.

Models of Analysis

We can illustrate the ideas of this section by focusing attention on the most interesting special case: general models of second-order number theory. Consider then the second-order language for number theory, with the parameters $\mathbf{0}, \mathbf{S}, <, \cdot,$ and $\mathbf{E}$. We take as our set of axioms the set A_E^2 obtained from A_E by adding as a twelfth member the Peano induction postulate (Example 2, Section 4.1). From Exercise 1 of Section 4.1, we can conclude that any model (in the semantics of that section) of A_E^2 is isomorphic to $\mathfrak{N}$.

But what of the *general* models of our axiom set? They can differ from $\mathfrak{N}$ in either (or both) of two ways. We can employ the compactness theorem as before to construct (nonstandard) general models of the axioms having infinite numbers (i.e., models $\mathfrak{A}$ with an element larger than the denotation of $\mathbf{S}^n\mathbf{0}$ in the ordering $<^{\mathfrak{A}}$). We can also find (non-absolute) general models in which, for example, the set universe (the unary relation universe) is less than the full power set of the individual universe. Indeed, any countable general model must be of this kind.

It is traditional for logicians to refer to second-order number theory as *analysis*. The name derives from the fact that it is possible to identify real numbers with sets of natural numbers. In second-order number theory we have quantifiers over sets of natural numbers, which we can view as quantifiers over real numbers. The appropriateness of the name is nevertheless open to question, but its usage is well established. By a *model of analysis* we will mean a general model of the above axiom set A_E^2.

Define an *ω-model of analysis* to be a model of analysis in which the individual universe is $\mathbb{N}$ and the denotations of $\mathbf{0}$ and S are the genuine 0 and S. (Consequently, the denotations of $<, +, \cdot,$ and $\mathbf{E}$ are also standard.) The motivation for studying ω-models can be stated as follows: We have a clear understanding — or so we think — of the set $\mathbb{N}$. But we do not have anything like the same understanding of its power set $\mathcal{P}\mathbb{N}$. For example, we may be uncertain whether its cardinality is $\aleph_1$ or $\aleph_2$ or more. So it is reasonable to hold fixed that which we are sure of ($\mathbb{N}$), but to leave open to interpretation by a structure that which we are not sure of ($\mathcal{P}\mathbb{N}$).

Among the ω-models of analysis there is the one *absolute* model, whose n-place relation universe consists of all n-ary relations on $\mathbb{N}$ (and whose function universes consist of all possible functions). A first-order sentence is true in an arbitrary ω-model of analysis iff it is true in $\mathfrak{N}$. But the ω-model may disagree with the absolute model on second-order sentences.

In the next theorem we assert that an ω-model of analysis is completely determined by its set universe (i.e., its one-place relation universe).

THEOREM 44B   If $\mathfrak{A}$ and $\mathfrak{B}$ are ω-models of analysis having the same one-place relation universe, then $\mathfrak{A} = \mathfrak{B}$.

PROOF.   Suppose R belongs to the three-place relation universe of $\mathfrak{A}$. Let $\langle R \rangle$ be the "compression" of R into a unary relation:

$$\langle R \rangle = \{\langle a, b, c \rangle \mid \langle a, b, c \rangle \in R\}.$$

Our sequence-encoding function is recursive and hence is first-order definable in number theory by a formula φ. $\langle R \rangle$ is in the set universe of $\mathfrak{A}$ by virtue of the comprehension sentence

$$\forall X^3 \, \exists X^1 \, \forall u [X^1 u \leftrightarrow \exists \, v_1 \, \exists \, v_2 \, \exists \, v_3 (\varphi(v_1, v_2, v_3, u) \\ \wedge X^3 v_1 v_2 v_3)].$$

Thus $\langle R \rangle$ is in the set universe of $\mathfrak{B}$; we unpack it by a similar argument. R is in the three-place relation universe of $\mathfrak{B}$ by virtue of the comprehension sentence

$$\forall X^1 \exists X^3 \, \forall \, v_1 \, \forall \, v_2 \, \forall \, v_3 [X^3 v_1 v_2 v_3 \leftrightarrow \\ \exists \, u(\varphi(v_1, v_2, v_3, u) \wedge X^1 u)].$$

A similar argument applies to the function universes. $\dashv$

Consequently, we can identify an ω-model of analysis with its set universe (which is included in $\mathcal{P}\mathbb{N}$). Not every subclass of $\mathcal{P}\mathbb{N}$ is then an ω-model of analysis, but only those for which the comprehension sentences are satisfied.

EXAMPLES OF ω-MODELS.   We need only specify the set universe.

1. $\mathcal{P}\mathbb{N}$ is the absolute model.

2. Let $(A; \in_A)$ be a model of the usual axioms for set theory such that (i) the relation $\in_A$ is the genuine membership relation $\{\langle a, b \rangle \mid a \in A, \ b \in A, \ \text{and} \ a \in b\}$ on the universe A, and (ii) A is transitive, i.e., if $a \in b \in A$, then $a \in A$. Then the collection of all those subsets of $\mathbb{N}$ that belong to A is an ω-model of analysis.

3. For a class $\mathcal{A} \subseteq \mathcal{P}\mathbb{N}$, define $\mathbb{D}\mathcal{A}$ to be the class of all sets $B \subseteq \mathbb{N}$ which are definable in the ω pre-structure with set universe $\mathcal{A}$ by a formula of the language of second-order number theory, augmented by parameters for each set in $\mathcal{A}$. Then define by transfinite recursion on the ordinals:

$$\begin{aligned} \mathcal{A}_0 &= \varnothing, \\ \mathcal{A}_{\alpha+1} &= \mathbb{D}\mathcal{A}_\alpha, \\ \mathcal{A}_\lambda &= \textstyle\bigcup_{\alpha < \lambda} \mathcal{A}_\alpha \quad \text{for limit } \lambda. \end{aligned}$$

By cardinality considerations we see that this stops growing at some ordinal β for which $\mathcal{A}_{\beta+1} = \mathcal{A}_\beta$. Let β_0 be the least

such β; it can be shown (from the Löwenheim–Skolem theorem) that β_0 is a countable ordinal. $\mathcal{A}_{\beta_0}$ coincides with $\bigcup_\alpha \mathcal{A}_\alpha$ (the union being over all ordinals α) and is called the class of *ramified analytical* sets. It is an ω-model of analysis; the truth of the comprehension sentences follows from the fact that $\mathbb{D}\mathcal{A}_{\beta_0} \subseteq \mathcal{A}_{\beta_0}$.

Suggestions for Further Reading

Jon Barwise (editor). *Handbook of Mathematical Logic*. North-Holland Publishing Company, Amsterdam, 1978. This "handbook" collects 31 expository articles on model theory, set theory, recursion theory, and proof theory, written by experts.

Jon Barwise and John Etchemendy. *The Language of First-Order Logic*. Center for the Study of Language and Information, Stanford, 1992. This introductory textbook comes with a disk for the *Tarski's World* software package. The same authors have produced the *Turing's World* and *Hyperproof* software packages.

J. L. Bell and M. Machover. *A Course in Mathematical Logic*. North-Holland Publishing Company, Amsterdam, 1977.

George Boolos and Richard Jeffrey. *Computability and Logic*. Cambridge University Press, Cambridge, 1974 (third edition 1989). This textbook gives a lively treatment of some of topics in the present book, addressed to a different audience.

C. C. Chang and H. J. Keisler. *Model Theory*. North-Holland Publishing Company, Amsterdam, 1973 (third edition 1990). This remains the classic book on model theory.

Herbert B. Enderton. *Elements of Set Theory*. Academic Press, New York, 1977. This is the author's favorite book on set theory.

Wilfrid Hodges. *A Shorter Model Theory*. Cambridge University Press, Cambridge, 1997. This is a shorter version of his *Model Theory*, published in 1993.

Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill Book Company, New York, 1967. This book is still the standard work in its field.

Joseph R. Shoenfield. *Mathematical Logic*. Association for Symbolic Logic and A K Peters, Natick, Massachusetts, 2000. First published by Addison-Wesley in 1967, the book gives a compact graduate-level treatment.

Jean van Heijenoort (editor). *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931.* Harvard University Press, Cambridge, Massachusetts, 1967. This is a collection of 46 fundamental papers in logic, translated into English and supplied with commentaries.

List of Symbols
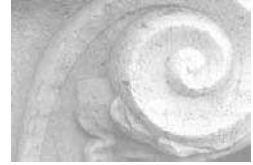
The numbers indicate the pages on which the symbol first occurs.

Symbol	Page	Symbol	Page	Symbol	Page
$\in$	1	$\dashv$	1	v_n	68
$\notin$	1	$\Rightarrow$	1	$=$	69
$=$	1	$\Leftarrow$	1	$\mathbf{0}$	70, 182
$A; t$	1	$\Leftrightarrow$	1	$\mathbf{S}$	71, 182
$\emptyset$	2	$\therefore$	1	$<$	70, 182
$\{x_1, \ldots, x_n\}$	2	$\not\models$	1	$+$	71, 182
$\{x \mid \_x\_\}$	2	$\neg$	11	$\cdot$	71, 182
$\mathbb{N}$	2	$\rightarrow$	11	$\mathbf{E}$	71, 182
$\mathbb{Z}$	2	$\wedge$	11	$\mathcal{F}_f$	74
$\subseteq$	2	$\vee$	12	$\mathcal{Q}_i$	75
$\mathcal{P}$	2	$\leftrightarrow$	14	$\neq$	1, 78
$\cup$	3	$\mathcal{E}$	17	$\not<$	78
$\cap$	3	F	20	$\lvert \mathfrak{A} \rvert$	81
$\bigcup$	3	T	20	c^{21}	81
$\bigcap$	3	$\overline{v}$	20	$\models_{\mathfrak{A}} \rho[s]$	83
$\langle x_1, \ldots, x_n \rangle$	4	$\models$	1	$\S$	83
$A \times B$	4	$\models=$	24	$s(x \mid d)$	84
$\operatorname{dom} R$	4	$\mathcal{D}$	32	$\models$	88
$\operatorname{ran} R$	4	$\#$	45	$\models\dashv$	88
$\operatorname{fld} R$	4	B_α^n	46	Mod	92
A^n	5	$\bot$	50	EC	92
$F : A \rightarrow B$	5	$\top$	50	EC_Δ	92
$[X]$	5	$\downarrow$	51	$\models_{\mathfrak{A}} \varphi[[a_1, \ldots, a_k]]$	90
$A \sim B$	8	$\vert$	51	$\mathfrak{A} \cong \mathfrak{B}$	94
$\operatorname{card} A$	8	$+$	51	$h \circ s$	96
$\preceq$	8	$\star$	62	$\mathfrak{A} \equiv \mathfrak{B}$	97
$\aleph$	9	$\forall$	68	$\mathbb{Q}$	98
$\mathbb{R}$	9	$\exists$	68, 77	$\forall_n$	00

Index