9

A Précis of First-Order Logic: Syntax

This chapter and the next contain a summary of material, mainly definitions, needed for later chapters, of a kind that can be found expounded more fully and at a more relaxed pace in introductory-level logic textbooks. Section 9.1 gives an overview of the two groups of notions from logical theory that will be of most concern: notions pertaining to formulas *and* sentences*, and notions pertaining to* truth under an interpretation*. The former group of notions, called* syntactic*, will be further studied in section 9.2, and the latter group, called* semantic*, in the next chapter.*

9.1 First-Order Logic

Logic has traditionally been concerned with relations among statements, and with properties of statements, that hold by virtue of 'form' alone, regardless of 'content'. For instance, consider the following argument:

(1) A mother or father of a person is an ancestor of that person.
(2) An ancestor of an ancestor of a person is an ancestor of that person.
(3) Sarah is the mother of Isaac, and Isaac is the father of Jacob.
(4) Therefore, Sarah is an ancestor of Jacob.

Logic teaches that the premisses (1)–(3) (*logically*) *imply* or have as a (*logical*) *consequence* the conclusion (4), because in any argument of the same form, if the premisses are true, then the conclusion is true. An example of another argument of the same form would be the following:

(5) A square or cube of a number is a power of that number.
(6) A power of a power of a number is a power of that number.
(7) Sixty-four is the cube of four and four is the square of two.
(8) Therefore, sixty-four is a power of two.

Modern logic represents the forms of statements by certain algebraic-looking symbolic expressions called *formulas*, involving special signs. The special signs we are going to be using are shown in Table 9-1.

Table 9-1. *Logical symbols*

$\sim$	Negation	'not . . . '
&	Conjunction	'. . . and . . . '
$\vee$	Disjunction	'. . . or . . . '
$\rightarrow$	Conditional	'if . . . then . . . '
$\leftrightarrow$	Biconditional	'. . . if and only if . . . '
$\forall x, \forall y, \forall z, \ldots$	Universal quantification	'for every x', 'for every y', 'for every z', . . .
$\exists x, \exists y, \exists z, \ldots$	Existential quantification	'for some x', 'for some y', 'for some z', . . .

In this symbolism, the form shared by the arguments (1)–(4) and (5)–(8) above might be represented as follows:

(9) $\forall x \forall y((\mathbf{P}yx \vee \mathbf{Q}yx) \rightarrow \mathbf{R}yx)$
(10) $\forall x \forall y(\exists z(\mathbf{R}yz \,\&\, \mathbf{R}zx) \rightarrow \mathbf{R}yx)$
(11) $\mathbf{Pab} \,\&\, \mathbf{Qbc}$
(12) $\mathbf{Rac}$

Content is put back into the forms by providing an *interpretation*. Specifying an interpretation involves specifying what sorts of things the xs and ys and zs are supposed to stand for, which of these things **a** and **b** and **c** are supposed to stand for, and which relations among these things **P** and **Q** and **R** are supposed to stand for. One interpretation would let the xs and ys and zs stand for (human) persons, **a** and **b** and **c** for the persons Sarah and Isaac and Jacob, and **P** and **Q** and **R** for the relations among persons of mother to child, father to child, and ancestor to descendent, respectively. With this interpretation, (9) and (10) would amount to the following more stilted versions of (1) and (2):

(13) For any person x and any person y, if either y is the mother of x or y is the father of x, then y is an ancestor of x.
(14) For any person x and any person y, if there is a person z such that y is an ancestor of z and z is an ancestor of x, then y is an ancestor of x.

(11) and (12) would amount to (3) and (4).

A different interpretation would let the xs and ys and zs stand for (natural) numbers, **a** and **b** and **c** for the numbers sixty-four and four and two, and **P** and **Q** and **R** for the relations of the cube or the square or a power of a number to that number, respectively. With this interpretation, (9)–(12) would amount to (5)–(8). We say that (9)–(11) imply (12) because in *any* interpretation in which (9)–(11) come out true, (12) comes out true.

Our goal in this chapter will be to make the notions of formula and interpretation rigorous and precise. In seeking the degree of clarity and explicitness that will be needed for our later work, the first notion we need is a division of the symbols that may occur in formulas into two sorts: *logical* and *nonlogical*. The logical symbols are the logical operators we listed above, the *connective symbols* (the tilde $\sim$, the ampersand &, the wedge $\vee$, the arrow $\rightarrow$, the double arrow $\leftrightarrow$), the *quantifier symbols* (the inverted ay $\forall$, the reversed ee $\exists$), plus the *variables* x, y, z, . . . that go with the quantifiers, plus left and right parentheses and commas for punctuation.

The nonlogical symbols are to begin with of two sorts: *constants* or *individual symbols*, and *predicates* or *relation symbols*. Each predicate comes with a fixed positive number of *places*. (It is possible to consider zero-place predicates, called *sentence letters*, but we have no need for them here.) As we were using them above, **a** and **b** and **c** were constants, and **P** and **Q** and **R** were two-place predicates.

Especially though not exclusively when dealing with mathematical material, some further apparatus is often necessary or useful. Hence we often include one more logical symbol, a special two-place predicate, the *identity symbol* or equals sign $=$, for '... is (the very same thing as) ...'. To repeat, the equals sign, though a two-place predicate, is counted as a logical symbol, but it is the only exception: all other predicates count as nonlogical symbols. Also, we often include one more category of nonlogical symbols, called *function symbols*. Each function symbol comes with a fixed number of *places*. (Occasionally, constants are regarded as zero-place function symbols, though usually we don't so regard them.)

We conscript the word 'language' to mean an enumerable set of nonlogical symbols. A special case is the *empty language* $L_\varnothing$, which is just the empty set under another name, with no nonlogical symbols. Here is another important case.

9.1 Example (The language of arithmetic). One language that will be of especial interest to us in later chapters is called the *language of arithmetic*, L^*. Its nonlogical symbols are the constant zero $\mathbf{0}$, the two-place predicate less-than $<$, the one-place function symbol successor $'$, and the two-place function symbols addition $+$ and multiplication $\cdot$.

Intuitively, *formulas* are just the sequences of symbols that correspond to grammatically well-formed sentences of English. Those that, like (9)–(12) above, correspond to English sentences that make a complete statement capable of being true or false are called *closed* formulas. Those that, like $(\mathbf{P}yz \vee \mathbf{Q}yx)$, correspond to English sentences involving unidentified xs and ys and zs that would have to be identified before the sentences could be said to be true or false, are called *open* formulas.

The *terms* are sequences of symbols, such as $\mathbf{0}$ or $\mathbf{0} + \mathbf{0}$ or x or x'', that correspond to grammatically well-formed phrases of English of the kind that grammarians call 'singular noun phrases'. The *closed* terms are the ones that involve no variables, and the *open* terms are the ones that involve variables whose values would have to be specified before the term as a whole could be said to have a denotation. When no function symbols are present, the only closed terms are constants, and the only open terms are variables. When function symbols are present, the closed terms also include such expressions as $\mathbf{0} + \mathbf{0}$, and the open terms such expressions as x''.

The formulas and terms of a given language are simply the ones all of whose nonlogical symbols belong to that language. Since languages are enumerable and each formula of a language is a finite string of symbols from the language plus variables and logical symbols, the set of formulas is enumerable, too. (One might at first guess that the empty language would have no formulas, but at least when identity is present, in fact it has infinitely many, among them $\forall x\, x = x$, $\forall y\, y = y$, $\forall z\, z = z$, and so on.)

An *interpretation* $\mathcal{M}$ for a language L consists of two components. On the one hand, there is a nonempty set $|\mathcal{M}|$ called the *domain* or *universe of discourse* of the

interpretation, the set of things $\mathcal{M}$ interprets the language to be talking about. When we say 'for every x' or 'for some x', what we mean, according to interpretation $\mathcal{M}$, is 'for every x in $|\mathcal{M}|$' or 'there exists an x in $|\mathcal{M}|$'. On the other hand, there is for each nonlogical symbol a *denotation* assigned to it. For a constant c, the denotation $c^{\mathcal{M}}$ is to be some individual in the domain $|\mathcal{M}|$. For an n-place nonlogical predicate R, the denotation $R^{\mathcal{M}}$ is to be some n-place relation on $|\mathcal{M}|$ (which is officially just a set of n-tuples of elements of $|\mathcal{M}|$, a one-place relation being simply a subset of $|\mathcal{M}|$).

For example, for the language L_G with constants **a** and **b** and **c** and two-place predicates **P** and **Q** and **R**, the genealogical interpretation $\mathcal{G}$ of L_G indicated above would now be described by saying that the domain $|\mathcal{G}|$ is the set of all persons, $\mathbf{a}^{\mathcal{G}}$ is Sarah, $\mathbf{b}^{\mathcal{G}}$ is Isaac, $\mathbf{c}^{\mathcal{G}}$ is Jacob, $\mathbf{P}^{\mathcal{G}}$ is set of ordered pairs of persons where the first is the mother of the second, and analogously for $\mathbf{Q}^{\mathcal{G}}$ and $\mathbf{R}^{\mathcal{G}}$. Under this interpretation, the open formula $\exists z(\mathbf{P}yz \,\&\, \mathbf{Q}zx)$ amounts to 'y is the paternal grandmother of x', while $\exists z(\mathbf{Q}yz \,\&\, \mathbf{P}zx)$ amounts to 'y is the maternal grandfather of x'. The closed formula $\sim\exists x\ \mathbf{P}xx$ amounts to 'no one is her own mother', which is true, while $\exists x\ \mathbf{Q}xx$ amounts to 'someone is his own father', which is false.

When the identity symbol is present, it is *not* treated like the other, nonlogical predicates: one is *not* free to assign it an arbitrary two-place relation on the domain as its denotation; rather, its denotation must be the genuine identity relation on that domain, the relation each thing bears to itself and to nothing else. When function symbols are present, for an n-place function symbol f, the denotation $f^{\mathcal{M}}$ is an n-argument function from $|\mathcal{M}|$ to $|\mathcal{M}|$.

9.2 Example (The standard interpretation of the language of arithmetic). One interpretation that will be of especial interest to us in later chapters is called the *standard interpretation* $\mathcal{N}^*$ of the language of the language of arithmetic L^*. Its domain $|\mathcal{N}^*|$ is the set of natural numbers; the denotation $\mathbf{0}^{\mathcal{N}^*}$ of the cipher **0** is the number zero; the denotation $<^{\mathcal{N}^*}$ of the less-than sign is the usual strict less-than order relation; the denotation $'^{\mathcal{N}^*}$ of the accent is the successor function, which takes each number to the next larger number; and the denotations $+^{\mathcal{N}^*}$ and $\cdot^{\mathcal{N}^*}$ of the plus sign and times sign are the usual addition and multiplication functions. Then such an open term as $x \cdot y$ would stand for the product of x and y, whatever they are; while such a closed term as $\mathbf{0}''$ would stand for the successor of the successor of zero, which is to say the successor of one, which is to say two. And such a closed formula as

(15) $\forall x\forall y(x \cdot y = \mathbf{0}'' \rightarrow (\ x = \mathbf{0}'' \vee y = \mathbf{0}''))$

would stand for 'for every x and every y, if the product of x and y is two, then either x is two or y is two' or 'a product is two only if one of the factors is two'. This happens to be true (given that our domain consists of natural numbers, with no negatives or fractions). Other closed formulas that come out true on this interpretation include the following:

(16) $\forall x\exists y(x < y \& \sim\exists z(x < z \& z < y))$

(17) $\forall x(x < x' \,\&\, \sim\exists z(x < z \& z < x'))$.

Here (16) says that for any number x there is a *next larger* number, and (17) that x' is precisely this next larger number.

(For the empty language $L_\varnothing$, there are no nonlogical symbols to be assigned denotations, but an interpretation must still specify a domain, and that specification makes a difference as to truth for closed formulas involving $=$. For instance, $\exists x \exists y \sim x = y$ will be true if the domain has at least two distinct elements, but false if it has only one.)

Closed formulas, which are also called *sentences*, have *truth values*, true or false, when supplied with an interpretation. But they may have different truth values under different interpretations. For our original example (9)–(12), on the genealogical interpretation we have since named $\mathcal{G}$ (and equally on the alternative arithmetical interpretation that we have left nameless) all four sentences came out true. But alternative interpretations are possible. For instance, if we kept everything else the same as in the genealogical interpretation, but took **R** to denote the relation of descendant to ancestor rather than vice versa, (10) and (11) would remain true, but (9) and (12) would become false: descendants of descendants are descendants, but parents and grandparents are not descendants. Various other combinations are possible. What one will *not* find is any interpretation that makes (9)–(11) all true, but (12) false. Precisely that, to repeat, is what is meant by saying that (9)–(11) *imply* (12).

9.3 Example (Alternative interpretations of the language of arithmetic). For the language of arithmetic, there is an alternative interpretation $\mathcal{Q}$ in which the domain is the nonnegative rational numbers, but the denotation of **0** is still zero, the denotation of $'$ is still the function that adds one to a number, the denotations of $+$ and $\cdot$ are the usual addition and multiplication operations, and the denotation of $<$ is still the less-than relation among the numbers in question. On this interpretation, (16) and (17) above are both false (because there are lots of rational numbers between x and any larger y in general, and lots of rational numbers between x and x plus one in particular). There is another alternative interpretation $\mathcal{P}$ in which the domain consists of the nonnegative half integers 0, $^1/_2$, 1, $1^1/_2$, 2, $2^1/_2$, 3, and so on, but the denotation of **0** is still zero, the denotation of $'$ is still the function that adds one to a number, the denotation of $+$ is still the usual addition operation, and the denotation of $<$ is still the less-than relation among the numbers in question. (Multiplication cannot be interpreted in the usual way, since a product of two half integers is not in general a half integer, but for purposes of this example it does not matter how multiplication is interpreted.) On this interpretation, (16) would be true (because there is no half integer between x and $y = x$ plus one-half), but (17) would be false (because there is a half integer between x and x plus one, namely x plus one-half). What you *won't* find is an interpretation that makes (17) true but (16) false. And again, that is what it means to say that (16) is a consequence of (17).

The explanations given so far provide part of the precision and rigor that will be needed in our later work, but only part. For they still rely on an intuitive understanding of what it is to be a sentence of a language, and what it is for a sentence be true in an interpretation. There are two reasons why we want to avoid this reliance on intuition. The first is that when we come to apply our work on computability to logic, we are going to want the notion of sentence to be so precisely defined that a *machine* could tell whether or not a given string of symbols is a sentence. The second is that

the notion of truth was historically under a certain cloud of suspicion, owing to the occurrence of certain contradictions, euphemistically called 'paradoxes', such as the ancient *Epimenides* or *liar* paradox: If I say, 'what I am now saying is not true', is what I am saying true? We are therefore going to want to give, for sentences of the kind of formal language we are considering, a definition of truth just as rigorous as the definition of any other notion in mathematics, making the notion of truth, as applied to the kind of formal language we are considering, as respectable as any other mathematical notion.

The next section will be devoted to giving precise and rigorous definitions of the notions of formula and sentence, and more generally to giving definitions of notions pertaining to *syntax*, that is, pertaining to the internal structure of formulas. The next chapter will be devoted to giving the definition of truth, and more generally to giving definitions of notions pertaining to *semantics*, that is, pertaining to the external interpretation of formulas.

9.2 Syntax

Officially we think of ourselves as working for each $k > 0$ with a fixed denumerable stock of k-place predicates:

$$\begin{array}{cccc} A_0^1 & A_1^1 & A_2^1 & \cdots \\ A_0^2 & A_1^2 & A_2^2 & \cdots \\ A_0^3 & A_1^3 & A_2^3 & \cdots \\ \vdots & \vdots & \vdots & \end{array}$$

and with a fixed denumerable stock of constants:

$$f_0^0 \qquad f_1^0 \qquad f_2^0 \qquad \ldots.$$

When function symbols are being used, we are also going to want for each $k > 0$ a fixed denumerable stock of k-place function symbols:

$$\begin{array}{cccc} f_0^1 & f_1^1 & f_2^1 & \cdots \\ f_0^2 & f_1^2 & f_2^2 & \cdots \\ f_0^3 & f_1^3 & f_2^3 & \cdots \\ \vdots & \vdots & \vdots & . \end{array}$$

Any language will be a subset of this fixed stock. (In some contexts in later chapters where we are working with a language L we will want to be able to assume that there are infinitely many constants available that have not been used in L. This is no real difficulty, even if L itself needs to contain infinitely many constants, since we can either add the new constants to our basic stock, or assume that L used only every other constant of our original stock to begin with.)

We also work with a fixed denumerable stock of variables:

$$v_0 \qquad v_1 \qquad v_2 \qquad \ldots.$$

Thus the more or less traditional $\mathbf{0}$ and $<$ and $'$ and $+$ and $\cdot$ we have been writing—and in practice, are going to continue to write—are in principle to be thought of as merely nicknames for f^0_0 and A^2_0 and f^1_0 and f^2_0 and f^2_1; while even writing x and y and z rather than v_i and v_j and v_k, we are using nicknames, too.

The official definition of the notion of formula begins by defining the notion of an *atomic formula*, which will be given first for the case where identity and function symbols are absent, then for the case where they are present. (If sentence letters were admitted, they would count as atomic formulas, too; but, as we have said, we generally are not going to admit them.) If identity and function symbols are absent, then an *atomic formula* is simply a string of symbols $R(t_1, \ldots, t_n)$ consisting of a predicate, followed by a left parenthesis, followed by n constants or variables, where n is the number of places of the predicate, with commas separating the successive terms, all followed by a right parenthesis. Further, if F is a formula, then so is its *negation* $\sim F$, consisting of a tilde followed by F. Also, if F and G are formulas, then so is their *conjunction* $(F \,\&\, G)$, consisting of a left parenthesis, followed by F, which is called the *left* or *first conjunct*, followed by the ampersand, followed by G, which is called the *right* or *second conjunct*, followed by a right parenthesis. Similarly for disjunction. Also, if F is a formula and x is a variable, the *universal quantification* $\forall xF$ is a formula, consisting of an inverted ay, followed by x, followed by F. Similarly for existential quantification.

And that is all: the definition of (*first-order*) *formula* is completed by saying that anything that is a (first-order) formula can be built up from atomic formulas in a sequence of finitely many steps—called a *formation sequence*—by applying negation, junctions, and quantifications to simpler formulas. (Until a much later chapter, where we consider what are called *second-order* formulas, 'first-order' will generally be omitted.)

Where identity is present, the atomic formulas will include ones of the kind $=(t_1, t_2)$. Where function symbols are present, we require a preliminary definition of terms. Variables and constants are *atomic* terms. If f is an n-place function symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term. And that is all: the definition of *term* is completed by stipulating that anything that is a term can be built up from atomic terms in a sequence of finitely many steps—called a *formation sequence*—by applying function symbols to simpler terms. Terms that contain variables are said to be *open*, while terms that do not are said to be *closed*. An atomic formula is now something of the type $R(t_1, \ldots, t_n)$ where the t_i may be any terms, not just constants or variables; but otherwise the definition of formula is unchanged.

Note that officially predicates are supposed to be written in front of the terms to which they apply, so writing $x < y$ rather than $<(x, y)$ is an unofficial colloquialism. We make use of several more such colloquialisms below. Thus we sometimes omit the parentheses around and commas separating terms in atomic formulas, and we generally write multiple conjunctions like $(A \,\&\, (B \,\&\, (C \,\&\, D)))$ simply as $(A \,\&\, B \,\&\, C \,\&\, D)$, and similarly for disjunctions, as well as sometimes omitting the outer parentheses on conjunctions and disjunctions $(F \,\&\, G)$ and $(F \vee G)$ when these stand alone rather than as parts of more complicated formulas. All this is slang, from the official point of view. Note that $\rightarrow$ and $\leftrightarrow$ have been left out of the official

Table 9-2. *Some terms of the language of arithmetic*

v_0	x
f_0^0	$\mathbf{0}$
$f_0^1(f_0^0)$	$\mathbf{1}$
$f_0^1(f_0^1(f_0^0))$	$\mathbf{2}$
$f_1^2(f_0^1(f_0^1(f_0^0)), v_0)$	$\mathbf{2} \cdot x$
$f_0^2(f_1^2(f_0^1(f_0^1(f_0^0)), v_0), f_1^2(f_0^1(f_0^1(f_0^0)), v_0))$	$\mathbf{2} \cdot x + \mathbf{2} \cdot x$

language entirely: $(F \rightarrow G)$ and $(F \leftrightarrow G)$ are to be considered unofficial abbreviations for $(\sim F \vee G)$ and $((\sim F \vee G)\ \&\ (\sim G \vee F))$. In connection with the language of arithmetic we allow ourselves two further such abbreviations, the bounded quantifiers $\forall y < x$ for $\forall y(y < x \rightarrow \ldots)$ and $\exists y < x$ for $\exists y(y < x\ \&\ \ldots)$.

Where identity is present, we also write $x = y$ and $x \neq y$ rather than $=(x, y)$ and $\sim=(x, y)$. Where function symbols are present, they also are supposed to be written in front of the terms to which they apply. So our writing x' rather than $'(x)$ and $x + y$ and $x \cdot y$ rather than $+(x, y)$ and $\cdot(x, y)$ is a colloquial departure from officialese. And if we adopt—as we do—the usual conventions of algebra that allow us to omit certain parenthesis, so that $x + y \cdot z$ is conventionally understood to mean $x + (y \cdot z)$ rather than $(x + y) \cdot z$ without our having to write the parentheses in explicitly, that is another such departure. And if we go further—as we do—and abbreviate $\mathbf{0}'$, $\mathbf{0}''$, $\mathbf{0}'''$, ..., as $\mathbf{1}$, $\mathbf{2}$, $\mathbf{3}$, ..., that is yet another departure.

Some terms of L^* in official and unofficial notation are shown in Table 9-2. The left column is a formation sequence for a fairly complex term.

Some formulas of L^* in official (or rather, semiofficial, since the the terms have been written colloquially) notation are shown in Table 9-3. The left column is a formation sequence for a fairly complex formula.

No one writing about anything, whether about family trees or natural numbers, will write in the official notation illustrated above (any more than anyone filling out a scholarship application or a tax return is going to do the necessary calculations in the rigid format established in our chapters on computability). The reader may well wonder why, if the official notation is so awkward, we don't just take the abbreviated

Table 9-3. *Some formulas of the language of arithmetic*

$A^2{}_0(x, \mathbf{0})$	$x < \mathbf{0}$
$A^2{}_0(x, \mathbf{1})$	$x < \mathbf{1}$
$A^2{}_0(x, \mathbf{2})$	$x < \mathbf{2}$
$A^2{}_0(x, \mathbf{3})$	$x < \mathbf{3}$
$\sim A^2{}_0(x, \mathbf{3}))$	$\sim x < \mathbf{3}$
$(= (x, \mathbf{1}) \vee = (x, \mathbf{2}))$	$x = \mathbf{1} \vee x = \mathbf{2}$
$(= (x, \mathbf{0}) \vee (= (x, \mathbf{1}) \vee = (x, \mathbf{2})))$	$x = \mathbf{0} \vee x = \mathbf{1} \vee x = \mathbf{2}$
$(\sim A^2{}_0(x, \mathbf{3}) \vee (= (x, \mathbf{0}) \vee (= (x, \mathbf{1}) \vee = (x, \mathbf{2}))))$	$x < \mathbf{3} \rightarrow (x = \mathbf{0} \vee x = \mathbf{1} \vee x = \mathbf{2})$
$\forall x(\sim A^2{}_0(x, \mathbf{3}) \vee (= (x, \mathbf{0}) \vee (= (x, \mathbf{1}) \vee = (x, \mathbf{2})))))$	$\forall x < \mathbf{3}(x = \mathbf{0} \vee x = \mathbf{1} \vee x = \mathbf{2})$

notation as the official one. The reason is that in proving things *about* the terms and formulas of a language, it is easiest if the language has a very rigid format (just as, in proving things *about* computability, it is easiest if the computations take place in a very rigid format). In writing examples of terms and formulas *in* the language, it is on the contrary easiest if the language has a very flexible format. The traditional strategy of logicians is to make the *official* language about which one proves theorems a very austere and rigid one, and to make the *unofficial* language in which one writes examples a very generous and flexible one. Of course, for the theorems proved about the austere idiom to be applicable to the generous idiom, one has to have confidence that all the abbreviations permitted by the latter but not the former *could in principle* be undone. But there is no need actually to undo them in practice.

The main method of proving theorems about terms and formulas in a language is called *induction on complexity*. We can prove that all formulas have a property by proving

Base Step: Atomic formulas have the property.

Induction Step: If a more complex formula is formed by applying a logical operator to a simpler formula or formulas, then, assuming (as *induction hypothesis*) that the simpler formula or formulas have the property, so does the more complex formula. The induction step will usually be divided into *cases*, according as the operator is $\sim$ or & or $\vee$ or $\forall$ or $\exists$.

Typically the proof will first be given for the situation where identity and function symbols are absent, then for the situation with identity present but function symbols absent, and then for the case with both identity and function symbols present. Identity typically requires very little extra work if any, but where function symbols are present, we generally need to prove some preliminary result about terms, which is also done by induction on complexity: we can prove that all terms have some property by proving that atomic terms have the property, and that if a more complex term is formed by applying a function symbol to simpler terms, then, assuming the simpler terms have the property, so does the more complex term.

The method of proof by induction on complexity is so important that we want to illustrate it now by very simple examples. The following lemma may tell us more than we want to know about punctuation, but is good practice.

9.4 Lemma (Parenthesis lemma). When formulas are written in official notation the following hold:

- **(a)** Every formula ends in a right parenthesis.
- **(b)** Every formula has equally many left and right parentheses.
- **(c)** If a formula is divided into a left part and a right part, both nonempty, then there are at least as many left as right parentheses in the left part, and more if that part contains at least one parenthesis.

Proof: We give first the proof for (a). *Base step:* An atomic formula $R(t_1, \ldots, t_n)$ or $=(t_1, t_2)$ of course ends in a right parenthesis. *Induction step, negation case:* If F ends in a right parenthesis, then so does $\sim F$, since the only new symbol is at the beginning. *Induction step, junction case:* A conjunction $(F \,\&\, G)$ or disjunction $(F \vee G)$ of course ends in a right parenthesis. *Induction step, quantification case:* If

F ends in a right parenthesis, then so do $\forall xF$ or $\exists xF$, for the same reason as in the case of negation, namely, that the only new symbols are at the beginning.

In giving the proof for (b), we allow ourselves to be a little less rigid about the format. We consider first the case where function symbols are absent. First note that an atomic formula $R(t_1, \ldots, t_n)$ or $=(t_1, t_2)$ has equal numbers of left and right parentheses, namely, one of each. Then note that F has equal numbers of left and right parentheses, then so does $\sim F$, since there are no new parentheses. Then note that if F has m of each kind of parenthesis, and G has n of each, then $(F \ \& \ G)$ has $m + n + 1$ of each, the only new ones being the outer ones. The proof for disjunction is the same as for conjunction, and the proofs for quantifications essentially the same as for negation.

If function symbols are present, we need the preliminary result that every term has equally many left and right parentheses. This is established by induction on complexity. An atomic term has equal numbers of left and right parentheses, namely zero of each. The nonatomic case resembles the conjunction case above: if s has m each of left and right parentheses, and t has n each, then $f(s, t)$ has $m + n + 1$ each; and similarly for $f(t_1, \ldots, t_k)$ for values of k other than two. Having this preliminary result, we must go back and reconsider the atomic case in the proof of (b). The argument now runs as follows: if s has m each of left and right parentheses, and t has n each, then $R(s, t)$ has $m + n + 1$ each, and similarly for $R(t_1, \ldots, t_k)$ for values of k other than two. No change is needed in the nonatomic cases of the proof of (b).

In giving the proof for (c), we also first consider the case where function symbols are absent. First suppose an atomic formula $R(t_1, \ldots, t_n)$ or $=(t_1, t_2)$ is divided into a left part λ and a right part ρ, both nonempty. If λ is just R or $=$, it contains zero parentheses of each kind. Otherwise, λ contains the one and only left parenthesis and not the one and only right parenthesis. In either case, (c) holds. Next assume (c) holds for F, and suppose $\sim F$ is divided. If λ consists just of $\sim$, and ρ of all of F, then λ contains zero parentheses of each kind. Otherwise, λ is of the form $\sim\lambda_0$, where λ_0 is a left part of F, and ρ is the right part of F. By assumption, then λ_0 and hence λ has at least as many left as right parentheses, and more if it contains any parentheses at all. Thus in all cases, (c) holds for $\sim F$. Next assume (c) holds for F and G, and suppose $(F \ \& \ G)$ is divided. The possible cases for the left part λ are:

Case1	Case 2	Case 3	Case 4	Case 5	Case 6
(	$(\lambda_0$	$(F$	$(F \ \&$	$(F \ \& \ \lambda_1$	$(F \ \& \ G$

where in case 2, λ_0 is a left part of F, and in case 5, λ_1 is a left part of G. In every case, the part of λ after the initial left parenthesis has at least as many left as right parentheses: obviously in case 1, by the assumption of (c) for F in case (2), by part (b) in case (3), and so on. So the whole left part λ has at least one more left than right parenthesis, and (c) holds for $(F \ \& \ G)$. The proof for disjunction is the same as for conjunction, and the proofs for quantifications essentially the same as for negation. We leave the case where function symbols are present to the reader.

We conclude this section with the official definitions of four more important syntactic notions. First, we officially define a string of consecutive symbols within a

given formula to be a *subformula* of the given formula if it is itself a formula. Where function symbols are present, we can similarly define a notion of *subterm*. We stop to note one result about subformulas.

9.5 Lemma (Unique readability lemma).

(a) The only subformula of an atomic formula $R(t_1, \ldots, t_n)$ or $=(t_1, t_2)$ is itself.
(b) The only subformulas of $\sim F$ are itself and the subformulas of F.
(c) The only subformulas of $(F \,\&\, G)$ or $(F \vee G)$ are itself and the subformulas of F and G.
(d) The only subformulas of $\forall x F$ or $\exists x F$ are itself and the subformulas of F.

These assertions may seem obvious, but they only hold because we use enough parentheses. If we used none at all, the disjunction of $F \,\&\, G$ with H, that is, $F \,\&\, G \vee H$, would have the subformula $G \vee H$, which is neither the whole conjunction nor a subformula of either conjunct. Indeed, the whole formula would be the same as the conjunction of F with $G \vee H$, and we would have a serious ambiguity. A rigorous proof of the unique readability lemma requires the parenthesis lemma.

Proof: For (a), a subformula of $R(t_1, \ldots, t_n)$ or $=(t_1, t_2)$ must contain the initial predicate R or $=$, and so, if it is not the whole formula, it will be a left part of it. Being a formula, it must contain (and in fact end in) a parenthesis by 9.4(a), and so, if it is not the whole formula but only a left part, must contain an excess of left over right parentheses by 9.4(c), which is impossible for a formula by 9.4(b).

For (b), a subformula of $\sim F$ that is not a subformula of F must contain the initial negation sign $\sim$, and so, if it is not the whole formula $\sim F$, it will be a left part of it, and from this point the argument is essentially the same as in the atomic case (a).

For (c), we relegate the proof to the problems at the end of the chapter.

For (d), the argument is essentially the same as for (b).

Resuming our series of definitions, second, using the notion of subformula, we state the official definition of which occurrences of a variable x in a formula F are *free* and which are *bound*: an occurrence of variable x is bound if it is part of a subformula beginning $\forall x \ldots$ or $\exists x \ldots$, in which case the quantifier $\forall$ or $\exists$ in question is said to *bind* that occurrence of the variable x, and otherwise the occurrence of the variable x is free. As an example, in

$$x < y \,\&\, \sim \exists z (x < z \,\&\, z < y)$$

all the occurrences of x and y are free, and all the occurrences of z are bound; while in

$$\mathbf{F}x \to \forall x \mathbf{F}x$$

the first occurrence of x is free, and the other two occurrences of x are bound. [The difference between the role of a free variable x and the role of a bound variable u in

a formula like $\forall u\ R(x, u)$ or $\exists u\ R(x, u)$ is not unlike the difference between the roles of x and of u in mathematical expressions like

$$\int_1^x \frac{\mathrm{du}}{u} \qquad \sum_{u=1}^{x} \frac{1}{u}$$

For some readers this analogy may be helpful, and those readers who do not find it so may ignore it.]

In general, any and all occurrences of variables in an atomic formula $R(t_1, \ldots, t_n)$ are free, since there are no quantifiers in the formula; the free occurrences of a variable in a negation $\sim F$ are just the free occurrences in F, since any subformula of $\sim F$ beginning $\forall x$ or $\exists x$ is a proper subformula of $\sim F$ and so a subformula of F; and similarly, the free occurrences of a variable in a junction $(F\ \&\ G)$ or $(F \vee G)$ are just those in F and G; and similarly, the free occurrences of a variable other than x in a quantification $\forall xF$ or $\exists xF$ are just those in F, while of course none of the occurrences of x in $\forall xF$ or $\exists xF$ is free.

Third, using the notion of free and bound occurrence of variables, we state the official definition of the notion of an *instance* of a formula. But before giving that definition, let us mention a convenient notational convention. When we write something like 'Let $F(x)$ be a formula', we are to be understood as meaning 'Let F be a formula in which no variables occur free except x'. That is, we indicate which variables occur free in the formula we are calling F by displaying them immediately after the name F we are using for that formula. Similarly, if we go on to write something like 'Let c be a constant, and consider $F(c)$', we are to be understood as meaning, 'Let c be a constant, and consider the result of substituting c for all free occurrences of x in the formula F'. That is, we indicate what substitution is to be made in the formula we are calling $F(x)$ by making that very substitution in the expression $F(x)$. Thus if $F(x)$ is $\forall y \sim y < x$, then $F(\mathbf{0})$ is $\forall y \sim y < \mathbf{0}$. Then the official definition of instance is just this: an *instance* of a formula $F(x)$ is any formula of form $F(t)$ for t a closed term. Similar notations apply where there is more than one free variable, and to terms as well as formulas.

Fourth and finally, again using the notion of free and bound occurrence of variables, we state the official definition of *sentence*: a formula is a sentence if no occurrence of any variable in it is free. A *subsentence* is a subformula that is a sentence.

Problems

9.1 Indicate the form of the following argument—traditionally called 'syllogism in Felapton'—using formulas:

(a) No centaurs are allowed to vote.

(b) All centaurs are intelligent beings.

(c) Therefore, some intelligent beings are not allowed to vote.

Do the premisses (a) and (b) in the preceding argument imply the conclusion (c)?

9.2 Consider (9)–(12) of at the beginning of the chapter, and give an alternative to the genealogical interpretation that makes (9) true, (10) false, (11) true, and (12) false.

9.3 Consider a language with a two-place predicate **P** and a one-place predicate **F**, and an interpretation in which the domain is the set of persons, the denotation of **P** is the relation of parent to child, and the denotation of **F** is the set of all female persons. What do the following amount to, in colloquial terms, under that interpretation?

(a) $\exists z \exists u \exists v (u \neq v \,\&\, \mathbf{P}uy \,\&\, \mathbf{P}vy \,\&\, \mathbf{P}uz \,\&\, \mathbf{P}vz \,\&\, \mathbf{P}zx \,\&\, \sim\mathbf{F}\, y)$

(b) $\exists z \exists u \exists v (u \neq v \,\&\, \mathbf{P}ux \,\&\, \mathbf{P}vx \,\&\, \mathbf{P}uz \,\&\, \mathbf{P}vz \,\&\, \mathbf{P}zy \,\&\, \mathbf{F}\, y)$

9.4 Officially, a *formation sequence* is a sequence of formulas in which each either is atomic, or is obtained by some earlier formula(s) in the sequence by negation, conjunction, disjunction, or universal or existential quantification. A formation sequence *for a formula* F is just a formation sequence whose last formula is F. Prove that in a formation sequence for a formula F, every subformula of F must appear.

9.5 Prove that every formula F has a formation sequence in which the *only* formulas that appear are subformulas of F, and the number of formulas that appear is no greater than the number of symbols in F.

9.6 Here is an outline of a proof that the only subformulas of $(F \,\&\, G)$ are itself and the subformulas of F and of G. Suppose H is some other kind of subformula. If H does not contain the displayed ampersand, then H must be of one of the two forms:

(a) (λ where λ is a left part of F, or

(b) ρ) where ρ is a right part of G.

If H does contain the displayed ampersand, then some subformula of H (possibly H itself) is a conjunction $(A \,\&\, B)$ where A and B are formulas and either

(c) $A = F$ and B is a left part λ of G,

(d) A is a right part ρ of F and $B = G$, or

(e) A is a right part ρ of F and B is a left part λ of G.

Show that (a) and (b) are impossible.

9.7 Continuing the preceding problem, show that (c)–(e) are all impossible.

9.8 Our definition allows the same variable to occur both bound and free in a formula, as in $P(x) \,\&\, \forall x Q(x)$. How could we change the definition to prevent this?

10

A Précis of First-Order Logic: Semantics

This chapter continues the summary of background material on logic needed for later chapters. Section 10.1 studies the notions of truth *and* satisfaction, *and section 10.2 the so-called* metalogical *notions of validity, implication or consequence, and (un)satisfiability.*

10.1 Semantics

Let us now turn from the official definitions of syntactical notions in the preceding chapter to the official definitions of semantic notions. The task must be to introduce the same level of precision and rigor into the definition of truth of a sentence in or on or under an interpretation as we have introduced into the notion of sentence itself. The definition we present is a version or variant of the *Tarski definition* of what it is for a sentence F to be true in an interpretation $\mathcal{M}$, written $\mathcal{M} \models F$. (The double turnstile $\models$ may be pronounced 'makes true'.)

The first step is to define truth for atomic sentences. The official definition will be given first for the case where identity and function symbols are absent, then for the case where they are present. (If sentence letters were admitted, they would be atomic sentences, and specifying which of them are true and which not would be part of specifying an interpretation; but, as we have said, we generally are not going to admit them.) Where identity and function symbols are absent, so that every atomic sentence has the form $R(t_1, \ldots, t_n)$ for some nonlogical predicate R and constants t_i, the definition is straightforward:

(1a) $$\mathcal{M} \models R(t_1, \ldots, t_n) \quad \text{if and only if} \quad R^{\mathcal{M}}\big(t_1^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}}\big).$$

The atomic sentence is true in the interpretation just in case the relation that the predicate is interpreted as denoting holds of the individuals that the constants are interpreted as denoting.

When identity is present, there is another kind of atomic sentence for which a definition of truth must be given:

(1b) $$\mathcal{M} \models {=}(t_1, t_2) \quad \text{if and only if} \quad t_1^{\mathcal{M}} = t_2^{\mathcal{M}}.$$

The atomic sentence is true in the interpretation just in case the individuals the constants are interpreted as denoting are the same.

When function symbols are present, we need a preliminary definition of the denotation $t^{\mathcal{M}}$ of a closed term t of a language L under an interpretation $\mathcal{M}$. Clauses (1a) and (1b) then apply, where the t_i may be any closed terms, and not just constants. For an atomic closed term, that is, for a constant c, specifying the denotation $c^{\mathcal{M}}$ of c is part of what is meant by specifying an interpretation. For more complex terms, we proceed as follows. If f is an n-place function symbol, then specifying the denotation $f^{\mathcal{M}}$ is again part of what is meant by specifying an interpretation. Suppose the denotations $t_1^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}}$ of terms $t_1, \ldots, t_n$ have been defined. Then we define the denotation of the complex term $f(t_1, \ldots, t_n)$ to be the value of the function $f^{\mathcal{M}}$ that is the denotation of f applied to the individuals $t_1^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}}$ that are the denotations of $t_1, \ldots, t_n$ as arguments:

(1c) $$(f(t_1, \ldots, t_n))^{\mathcal{M}} = f^{\mathcal{M}}\big(t_1^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}}\big).$$

Since every term is built up from constants by applying function symbols a finite number of times, these specifications determine the denotation of every term.

So, for example, in the standard interpretation of the language of arithmetic, since **0** denotes the number zero and $'$ denotes the successor function, according to (1c) $\mathbf{0'}$ denotes the value obtained on applying the successor function to zero as argument, which is to say the number one, a fact we have anticipated in abbreviating $\mathbf{0'}$ as **1**. Likewise, the denotation of $\mathbf{0''}$ is the value obtained on applying the successor function to the denotation of $\mathbf{0'}$, namely one, as argument, and this value is of course the number two, again a fact we have been anticipating in abbreviating $\mathbf{0''}$ as **2**. Similarly, the denotation of $\mathbf{0'''}$ is three, as is, for instance, the denotation of $\mathbf{0' + 0''}$. No surprises here.

According to (1b), continuing the example, since the denotations of $\mathbf{0'''}$ or **3** and of $\mathbf{0' + 0''}$ or $\mathbf{1 + 2}$ are the same, $\mathbf{0''' = 0' + 0''}$ or $\mathbf{3 = 1 + 2}$ is true, while by contrast $\mathbf{0'' = 0' + 0''}$ or $\mathbf{2 = 1 + 2}$ is false. Again no surprises. According to (1a), further continuing the example, since the denotation of $<$ is the strict less-than relation, and the denotations of $\mathbf{0'''}$ or **3** and of $\mathbf{0' + 0''}$ or $\mathbf{1 + 2}$ are both three, the atomic sentence $\mathbf{0''' < 0' + 0''}$ or $\mathbf{3 < 1 + 2}$ is false, while by contrast $\mathbf{0'' < 0' + 0''}$ is true. Yet again, no surprises.

There is only one candidate for what the definition should be in each of the cases of negation and of the two junctions:

(2a) $\mathcal{M} \models \sim F$ if and only if not $\mathcal{M} \models F$

(2b) $\mathcal{M} \models (F \,\&\, G)$ if and only if $\mathcal{M} \models F$ and $\mathcal{M} \models G$

(2c) $\mathcal{M} \models (F \vee G)$ if and only if $\mathcal{M} \models F$ or $\mathcal{M} \models G$.

So, for example, in the standard interpretation of the language of arithmetic, since $\mathbf{0 = 0}$ and $\mathbf{0 < 0'}$ are true while $\mathbf{0 < 0}$ is false, we have that $(\mathbf{0 = 0} \vee \mathbf{0 < 0'})$ is true, $(\mathbf{0 < 0} \,\&\, \mathbf{0 = 0})$ is false, $(\mathbf{0 < 0} \,\&\, (\mathbf{0 = 0} \vee \mathbf{0 < 0'}))$ is false, and $((\mathbf{0 < 0} \,\&\, \mathbf{0 = 0}) \vee \mathbf{0 < 0'})$ is true. Still no surprises.

One consequence of (2a)–(2c) worth mentioning is that $(F \,\&\, G)$ is true if and only if $\sim(\sim F \vee \sim G)$ is true, and $(F \vee G)$ is true if and only if $\sim(\sim F \,\&\, \sim G)$ is true. We could therefore if we wished drop one of the pair &, $\vee$ from the official language, and treat it as an unofficial abbreviation (for an expression involving $\sim$ and the other of the pair) on a par with $\rightarrow$ and $\leftrightarrow$.

The only slight subtlety in the business arises at the level of quantification. Here is a simple, tempting, and *wrong* approach to defining truth for the case of quantification, called the *substitutional* approach:

$$\begin{array}{lll} \mathcal{M} \models \forall x\, F(x) & \text{if and only if} & \text{for every closed term } t,\ \mathcal{M} \models F(t) \\ \mathcal{M} \models \exists x\, F(x) & \text{if and only if} & \text{for some closed term } t,\ \mathcal{M} \models F(t). \end{array}$$

In other words, under this definition a universal quantification is true if and only if every substitution instance is true, and an existential quantification is true if and only if some substitution instance is true. This definition in general produces results not in agreement with intuition, unless it happens that every individual in the domain of the interpretation is denoted by some term of the language. If the domain of the interpretation is enumerable, we could always *expand* the language to add more constants and extend the interpretation so that each individual in the domain is the denotation of one of them. But we cannot do this when the domain is nonenumerable. (At least we cannot do so while continuing to insist that a language is supposed to involve only a finite or enumerable set of symbols. Of course, to allow a 'language' with a nonenumerable set of symbols would involve a considerable stretching of the concept. We will briefly consider this extended concept of 'language' in a later chapter, but for the moment we set it aside.)

10.1 Example. Consider the language L^* of arithmetic and three different interpretations of it: first, the standard interpretation $\mathcal{N}^*$; second, the alternative interpretation $\mathcal{Q}$ we considered earlier, with domain the nonnegative rational numbers; third, the similar alternative interpretation $\mathcal{R}$ with domain the nonnegative real numbers. Now in fact the substitutional approach gives the intuitively correct results for $\mathcal{N}^*$ in all cases. Not so, however, for the other two interpretations. For, all closed terms in the language have the same denotation in all three interpretations, and from this it follows that all closed terms denote natural numbers. And from this it follows that $t + t = \mathbf{1}$ is false for all closed terms t, since there is no natural number that, added to itself, yields one. So on the substitutional approach, $\exists x(x + x = \mathbf{1})$ would come out false on all three interpretations. But intuitively 'there is something (in the domain) that added to itself yields one' is false only on the standard interpretation $\mathcal{N}^*$, and true on the rational and real interpretations $\mathcal{Q}$ and $\mathcal{R}$.

We could try to fix this by adding more constants to the language, so that there is one denoting each nonnegative rational number. If this were done, then on the rational and real interpretations, $\mathbf{1/2} + \mathbf{1/2} = \mathbf{1}$ would come out true, and hence $\exists x(x + x = \mathbf{1})$ would come out true using the substitutional approach, and this particular example of a problem with the substitutional approach would be fixed. Indeed, the substitutional approach would then give the intuitively correct results for $\mathcal{Q}$ in all cases. Not so, however, for $\mathcal{R}$. For, all terms in the language would denote rational numbers, and from this it would follow that $t \cdot t = \mathbf{2}$ is false for all terms t, since there is no rational number that, multiplied by itself,

yields two. So on the substitutional approach, $\exists x(x \cdot x = \mathbf{2})$ would come out false. But intuitively, though 'there is something (in the domain) that multiplied by itself yields two' is false on the rational interpretation, it is true on the real interpretation. We could try to fix this by adding yet more terms to the language, but by Cantor's theorem there are too many real numbers to add a term for each of them while keeping the language enumerable.

The *right* definition for the case of quantification has to be a little more indirect. In defining when $\mathcal{M} \models \forall x F(x)$ we do not attempt to extend the given language L so as to provide constants for every individual in the domain of the interpretation at once. In general, that cannot be done without making the language nonenumerable. However, if we consider any particular individual in the domain, we *could* extend the language and interpretation to give *just it* a name, and what we do in defining when $\mathcal{M} \models \forall x F(x)$ is to consider *all possible* extensions of the language and interpretation by adding just one new constant and assigning it a denotation.

Let us say that in the interpretation $\mathcal{M}$ the individual m *satisfies* $F(x)$, and write $\mathcal{M} \models F[m]$, to mean 'if we considered the extended language $L \cup \{c\}$ obtained by adding a new constant c in to our given language L, and if among all the extensions of our given interpretation $\mathcal{M}$ to an interpretation of this extended language we considered the one $\mathcal{M}^c_m$ that assigns c the denotation m, then $F(c)$ would be true':

(3*) $\qquad \mathcal{M} \models F[m] \quad$ if and only if $\quad \mathcal{M}^c_m \models F(c).$

(For definiteness, let us say the constant to be added should be the first constant not in L in our fixed enumeration of the stock of constants.)

For example, if $F(x)$ is $x \cdot x = \mathbf{2}$, then on the real interpretation of the language of arithmetic $\sqrt{2}$ satisfies $F(x)$, because if we extended the language by adding a constant c and extended the interpretation by taking c to denote $\sqrt{2}$, then $c \cdot c = \mathbf{2}$ would be true, because the real number denoted by c would be one that, multiplied by itself, yields two. This definition of satisfaction can be extended to formulas with more than one free variable. For instance, if $F(x, y, z)$ is $x \cdot y = z$, then $\sqrt{2}, \sqrt{3}, \sqrt{6}$ satisfy $F(x, y, z)$, because if we added c, d, e denoting them, $c \cdot d = e$ would be true.

Here, then, is the *right* definition, called the *objectual* approach:

(3a) $\quad \mathcal{M} \models \forall x F(x) \quad$ if and only if $\quad$ for every m in the domain, $\mathcal{M} \models F[m]$

(3b) $\quad \mathcal{M} \models \exists x F(x) \quad$ if and only if $\quad$ for some m in the domain, $\mathcal{M} \models F[m]$.

So $\mathcal{R} \models \exists x F(x)$ under the above definitions, in agreement with intuition, even though there is no term t in the actual language such that $\mathcal{R} \models F(t)$, because $\mathcal{R} \models F[\sqrt{2}]$.

One immediate implication of the above definitions worth mentioning is that $\forall x F$ turns out to be true just in case $\sim\exists x \sim F$ is true, and $\exists x F$ turns out to be true just in case $\sim\forall x \sim F$ is true, so it would be possible to drop one of the pair $\forall, \exists$ from the official language, and treat it as an unofficial abbreviation.

The method of proof by induction on complexity can be used to prove semantic as well as syntactic results. The following result can serve as a warm-up for more substantial proofs later, and provides an occasion to review the definition of truth clause by clause.

10.2 Proposition (Extensionality lemma).

(a) Whether a sentence A is true depends only on the domain and denotations of the nonlogical symbols in A.

(b) Whether a formula $F(x)$ is satisfied by an element m of the domain depends only on the domain, the denotations of the nonlogical symbols in F, and the element m.

(c) Whether a sentence $F(t)$ is true depends only on the domain, the denotations of the nonlogical symbols in $F(x)$, and the denotation of the closed term t.

Here (a), for instance, means that the truth value of A does not depend on what the nonlogical symbols in A *themselves* are, but only on what their *denotations* are, and does not depend on the denotations of nonlogical symbols *not* in A. (So a more formal statement would be: If we start with a sentence A and interpretation I, and change A to B by changing zero or more nonlogical symbols to others of the same kind, and change I to $\mathcal{J}$, then the truth value of B in $\mathcal{J}$ will be the same as the truth value of A in I provided $\mathcal{J}$ has the same domain as I, $\mathcal{J}$ assigns each unchanged nonlogical symbol the same denotation I did, and whenever a nonlogical symbol S is changed to T, then $\mathcal{J}$ assigns to T the same denotation I assigned to S. The proof, as will be seen, is hardly longer than this formal statement!)

Proof: In proving (a) we consider first the case where function symbols are absent, so the only closed terms are constants, and proceed by induction on complexity. By the atomic clause in the definition of truth, the truth value of an atomic sentence depends only on the denotation of the predicate in it (which in the case of the identity predicate cannot be changed) and the denotations of the constants in it. For a negation $\sim B$, assuming as induction hypothesis that (a) holds for B, then (a) holds for $\sim B$ as well, since by the negation clause in the definition of truth, the truth value of $\sim B$ depends only on the truth value of B. The cases of disjunction and conjunction are similar.

For a universal quantification $\forall x\, B(x)$, assuming as induction hypothesis that (a) holds for sentences of form $B(c)$, then (b) holds for $B(x)$, for the following reason. By the definition of satisfaction, whether m satisfies $B(x)$ depends on the truth value of $B(c)$ where c is a constant not in $B(x)$ that is assigned denotation m. [For definiteness, we specified which constant was to be used, but the assumption of (a) for sentences of form $B(c)$ implies that it does not matter what constant is used, so long as it is assigned denotation m.] By the induction hypothesis, the truth value of $B(c)$ depends only on the domain and the denotations of the nonlogical symbols in $B(c)$, which is to say, the denotations of the nonlogical symbols in $B(x)$ and the element m that is the denotation of the nonlogical symbol c, just as asserted by (b) for $B(x)$. This preliminary observation made, (a) for $\forall x\, B(x)$ follows at once, since by the universal quantification clause in the definition of truth, the truth value of $\forall x\, B(x)$ depends only on the domain and which of its elements satisfy $B(x)$. The case of existential quantification is the same.

If function symbols are present, we must as a preliminary establish by induction on complexity of terms that the denotation of a term depends only on the denotations of the nonlogical symbols occurring in it. This is trivial in the case of a constant. If it

is true for terms $t_1, \ldots, t_n$, then it is true for the term $f(t_1, \ldots, t_n)$, since the definition of denotation of term mentions only the denotation of the nonlogical symbol f and the denotations of the terms $t_1, \ldots, t_n$. This preliminary observation made, (a) for atomic sentences follows, since by the atomic clause in the definition of truth, the truth value of an atomic sentence depends only on the denotation of its predicate and the denotations of its terms. The nonatomic cases in the proof require no change.

We have proved (b) in the course of proving (a). Having (b), the proof of (c) reduces to showing that whether a sentence $F(t)$ is true depends only on whether the element m denoted by t satisfies $F(x)$, which by the definition of satisfaction is to say, on whether $F(c)$ is true, where c is a constant having the same denotation m as t. The proof that $F(c)$ and $F(t)$ have the same truth value if c and t have the same denotation is relegated to the problems at the end of the chapter.

It is also extensionality (specifically, part (c) of Proposition 10.2) that justifies our earlier passing remarks to the effect that the substitutional approach to defining quantification does work *when every element of the domain is the denotation of some closed term.* If for some closed term t the sentence $B(t)$ is true, then letting m be the denotation of t, it follows by extensionality that m satisfies $B(x)$, and hence $\exists x\, B(x)$ is true; and conversely, if $\exists x\, B(x)$ is true, then some m satisfies $B(x)$, and *assuming that every element of the domain is the denotation of some closed term*, then some term t denotes m, and by extensionality, $B(t)$ is true. Thus under the indicated assumption, $\exists x B(x)$ is true if and only if for some term t, $B(t)$ is true, and similarly $\forall x\, B(x)$ is true if and only if for every term t, $B(t)$ is true.

Similarly, if every element of the domain is the denotation of a closed term *of some special kind* then $\exists x\, B(x)$ is true if and only if $B(t)$ is true for some closed term t that is *of that special kind*. In particular, for the standard interpretation $\mathcal{N}^*$ of the language of arithmetic L^*, where every element of the domain is the denotation of one of the terms $\mathbf{0}$, $\mathbf{1}$, $\mathbf{2}$, ..., we have

$$\mathcal{N}^* \models \forall x\, F(x) \quad \text{if and only if} \quad \text{for every natural number } m, \mathcal{N}^* \models F(\mathbf{m})$$
$$\mathcal{N}^* \models \exists x\, F(x) \quad \text{if and only if} \quad \text{for some natural number } m, \mathcal{N}^* \models F(\mathbf{m})$$

where $\mathbf{m}$ is the numeral for the number m (that is, the term consisting of the cipher $\mathbf{0}$ followed by m copies of the accent $'$).

10.2 Metalogical Notions

Now that rigorous definitions of formula and sentence, and of satisfaction and truth, have been given, we can proceed to the definitions of the main notions of logical theory. A set of sentences Γ *implies* or has as a *consequence* the sentence D if there is no interpretation that makes every sentence in Γ true, but makes D false. This is the same as saying that every interpretation that makes every sentence in Γ true makes D true. (Or almost the same. Actually, if D contains a nonlogical symbol not in Γ, an interpretation might make Γ true but assign no denotation to this symbol and therefore no truth value to D. But in such a case, however the denotation is extended to assign a denotation to any such symbols and therewith a truth value to D, Γ will

still be true by the extensionality lemma, so D cannot be false and must be true. To avoid fuss over such points, in future we tacitly understand 'every interpretation' to mean 'every interpretation that assigns denotations to all the nonlogical symbols in whatever sentences we are considering'.) We use 'makes every sentence in Γ true' and 'makes Γ true' interchangeably, and likewise 'the sentences in the set Γ imply D' and 'Γ implies D'. When Γ contains but a single sentence C (in symbols, when $\Gamma = \{C\}$), we use 'Γ implies D' and 'C implies D' interchangeably. Let us give a few examples. There are more in the problems at the end of the chapter (and many, many, many more in introductory textbooks).

10.3 Example. Some implication principles

(a) $\sim\sim B$ implies B.

(b) B implies $(B \vee C)$ and C implies $(B \vee C)$.

(c) $\sim(B \vee C)$ implies $\sim B$ and $\sim C$.

(d) $B(t)$ implies $\exists x\, B(x)$.

(e) $\sim\exists x\, B(x)$ implies $\sim B(t)$.

(f) $s = t$ and $B(s)$ imply $B(t)$.

Proofs: For (a), by the negation clause in the definition of truth, in any interpretation, if $\sim\sim B$ is true, then $\sim B$ must be false, and B must be true. For (b), by the disjunction clause in the definition of truth, in any interpretation, if B is true, then $(B \vee C)$ is true; similarly for C. For (c), by what we have just shown, any interpretation that does *not* make $(B \vee C)$ true can*not* make B true; hence any intepretation that makes $\sim(B \vee C)$ true makes $\sim B$ true; and similarly for $\sim C$. For (d), in any interpretation, by the extensionality lemma $B(t)$ is true if and only if the element m of the domain that is denoted by t satisfies $B(x)$, in which case $\exists x\, B(x)$ is true. As for (e), it follows from what we have just shown much as (c) follows from (b). For (f), by the identity clause in the definition of truth, in any interpretation, if $s = t$ is true, then s and t denote the same element of the domain. Then by the extensionality lemma $B(s)$ is true if and only if $B(t)$ is true.

There are two more important notions to go with implication or consequence. A sentence D is *valid* if no interpretation makes D false. In this case, *a fortiori* no interpretation makes Γ true and D false; Γ implies D for *any* Γ. Conversely, if every Γ implies D, then since for every interpretation there is a set of sentences Γ it makes true, no interpretation can make D false, and D is valid. A set of sentences Γ is *unsatisfiable* if no interpretation makes Γ true (and is *satisfiable* if some interpretation does). In this case, *a fortiori* no interpretation makes Γ true and D false, so Γ implies D for any D. Conversely, if Γ implies every D, then since for every interpretation there is a sentence it makes false, there can be no interpretation making Γ true, and Γ is unsatisfiable.

Notions such as consequence, unsatisfiability, and validity are often called 'metalogical' in contrast to the notions of negation, conjunction, disjunction, and universal and existential quantification, which are simply called 'logical'. Terminology aside, the difference is that there are symbols $\sim$, &, $\vee$, $\forall$, $\exists$ in our formal language

(the 'object language') for negation and the rest, whereas words like 'consequence' only appear in the unformalized prose, the mathematical English, in which we talk *about* the formal language (the 'metalanguage').

Just as for implication or consequence, so for validity and for unsatisfiability and satisfiability, there are innumerable little principles that follow directly from the definitions. For instance: if a set is satisfiable, then so is every subset (since an interpretation making every sentence in the set true will make every sentence in the subset true); no set containing both a sentence and its negation is satisfiable (since no interpretation makes them both true); and so on. The plain assertions of Example 10.3 can each be elaborated into fancier versions about validity and (un)satisfiability, as we next illustrate in the case of 10.3(a).

10.4 Example. Variations on a theme

(a) $\sim\sim B$ implies B.

(b) If Γ implies $\sim\sim B$, then Γ implies B.

(c) If B implies D, then $\sim\sim B$ implies D.

(d) If $\Gamma \cup \{B\}$ implies D, then $\Gamma \cup \{\sim\sim B\}$ implies D.

(e) If $\sim\sim B$ is valid, then B is valid.

(f) If $\Gamma \cup \{B\}$ is unsatisfiable, then $\Gamma \cup \{\sim\sim B\}$ is unsatisfiable.

(g) If $\Gamma \cup \{\sim\sim B\}$ is satisfiable, then $\Gamma \cup \{B\}$ is satisfiable.

Proof: (a) is a restatement of 10.3(a). For (b), we are given that every interpretation that makes Γ true makes $\sim\sim B$ true, and want to show that any interpretation that makes Γ true makes B true. But this is immediate from (a), which says that any interpretation that makes $\sim\sim B$ true makes B true. For (c), we are given that any interpretation that makes B true makes D true, and want to show that any interpretation that makes $\sim\sim B$ true makes D true. But again, this is immediate from the fact that any interpretation that makes $\sim\sim B$ true makes B true. In (d), $\Gamma \cup \{B\}$ denotes the result of adding B to Γ. The proof in this case is a combination of the proofs of (b) and (c). For (e), we are given that every interpretation makes $\sim\sim B$ true, and want to show that every interpretation makes B true, while for (f), we are given that no interpretation makes Γ and B true, and want to show that no interpretation makes Γ and $\sim\sim B$ true. But again both are immediate from (a), that is, from the fact that every interpretation that makes $\sim\sim B$ true makes B true. Finally, (g) is immediate from (f).

We could play the same game with any of 10.3(b)–10.3(f). Some results exist only in the fancy versions, so to speak.

10.5 Example. Some satisfiability principles

(a) If $\Gamma \cup \{(A \vee B)\}$ is satisfiable, then either $\Gamma \cup \{A\}$ is satisfiable, or $\Gamma \cup \{B\}$ is satisfiable.

(b) If $\Gamma \cup \{\exists x B(x)\}$ is satisfiable, then for any constant c not occurring in Γ or $\exists x B(x)$, $\Gamma \cup \{B(c)\}$ is satisfiable.

(c) If Γ is satisfiable, then $\Gamma \cup \{t = t\}$ is satisfiable.

Proof: For (a), we are given that some interpretation makes Γ and $A \vee B$ true, and want to show that some interpretation makes Γ and A true, or some makes Γ and B true. In fact, the *same* interpretation that makes Γ and $A \vee B$ true either makes A true or makes B true, by the disjunction clause in the definition of truth. For (b), we are given that some interpretation makes Γ and $\exists x\, B(x)$ true, and want to show that some interpretation makes Γ and $B(c)$ true, assuming c does not occur in Γ or $\exists x\, B(x)$. Well, since $\exists x\, B(x)$ is true, some element m of the domain satisfies $B(x)$. And since c does not occur in Γ or $\exists x\, B(x)$, we can change the interpretation to make m the denotation of c, without changing the denotations of any nonlogical symbols in Γ or $\exists x\, B(x)$, and so by extensionality not changing their truth values. But then Γ is still true, and since m satisfies $B(x)$, $B(c)$ is also true. For (c), we are given that some interpretation makes Γ true and want to show that some interpretation makes Γ and $t = t$ true. But *any* interpretation makes $t = t$ true, so long as it assigns a denotation to each nonlogical symbol in t, and if our given interpretation does not, it at least assigns a denotation to every nonlogical symbol in t that occurs in Γ, and if we extend it to assign denotations to any other nonlogical symbols in t, by extensionality Γ will still be true, and now $t = t$ will be true also.

There is one more important metalogical notion: two sentences are *equivalent over an interpretation* $\mathcal{M}$ if they have the same truth value. Two formulas $F(x)$ and $G(x)$ are equivalent over $\mathcal{M}$ if, taking a constant c occurring in neither, the sentences $F(c)$ and $G(c)$ are equivalent over every interpretation $\mathcal{M}^c_m$ obtained by extending $\mathcal{M}$ to provide some denotation m for c. Two sentences are (*logically*) *equivalent* if they are equivalent over all interpretations. Two formulas $F(x)$ and $G(x)$ are (*logically*) equivalent if, taking a constant c occurring in neither, the sentences $F(c)$ and $G(c)$ are (logically) equivalent. A little thought shows that formulas are (logically) equivalent if they are equivalent over every interpretation. The definitions may be extended to formulas with more than one free variable. We leave the development of the basic properties of equivalence entirely to the problems.

Before closing this chapter and bringing on those problems, a remark will be in order. The method of induction on complexity we have used in this chapter and the preceding to prove such unexciting results as the parenthesis and extensionality lemmas will eventually be used to prove some less obvious and more interesting results. Much of the interest of such results about formal languages depends on their being applicable to ordinary language. We have been concerned here mainly with how to read sentences of our formal language in ordinary language, and much less with writing sentences of ordinary language in our formal language, so we need to say a word about the latter topic.

In later chapters of this book there will be many examples of writing assertions from *number theory*, the branch of mathematics concerned with the natural numbers, as first-order sentences in the language of arithmetic. But the full scope of what can be done with first-order languages will not be apparent from these examples, or this book, alone. Works on set theory give examples of writing assertions from other branches of mathematics as first-order sentences in a *language of set theory*, and make it plausible that in virtually *all* branches of mathematics, what we want to say

can be said in a first-order language. Works on logic at the introductory level contain a wealth of examples of how to say what we want to say in a first-order language from outside mathematics (as in our genealogical examples).

But this cannot *always* be done outside of mathematics, and some of our results *do not* apply unrestrictedly to ordinary language. A case in point is unique readability. In ordinary language, ambiguous sentences of the type 'A and B or C' are perfectly possible. Of course, though *possible*, they are not *desirable*: the sentence ought to be rewritten to indicate whether 'A, and either B or C' or 'Either A and B, or C' is meant. A more serious case in point is extensionality. In ordinary language it is *not* always the case that one expression can be changed to another denoting the same thing without altering truth values. To give the classic example, Sir Walter Scott was the author of the historical novel *Waverley*, but there was a time when this fact was unknown, since the work was originally published anonymously. At that time, 'It is known that Scott is Scott' was as always true, but 'It is known that the author of *Waverley* is Scott' was false, even though 'Scott' and 'the author of *Waverly*' had the same denotation.

To put the matter another way, writing s for 'Scott' and t for 'the author of *Waverley*', and writing $A(x)$ for 'x is Scott' and $\Box$ for 'it is known that', what we have just said is that $s = t$ and $\Box A(s)$ may be true without $\Box A(t)$ being true, in contrast to one of our examples above, according to which, in our formal languages, $s = t$ and $B(s)$ always imply $B(t)$. There is no contradiction with our example, of course, since our formal languages do not contain any operator like $\Box$; but for precisely this reason, *not* everything that can be expressed in ordinary language can be expressed in our formal languages. There is a separate branch of logic, called *modal logic*, devoted to operators like $\Box$, and we are eventually going to get a peek at a corner of this branch of logic, though only in the last chapter of the book.

Problems

10.1 Complete the proof of the extensionality lemma (Proposition 10.2) by showing that if c is a constant and t a closed term having the same denotation, then substituting t for c in a sentence does not change the truth value of the sentence.

10.2 Show that $\exists y \forall x R(x, y)$ implies $\forall x \exists y R(x, y)$.

10.3 Show that $\forall x \exists y F(x, y)$ does not imply $\exists y \forall x F(x, y)$.

10.4 Show that:

(a) If the sentence E is implied by the set of sentences Δ and every sentence D in Δ is implied by the set of sentences Γ, then E is implied by Γ.

(b) If the sentence E is implied by the set of sentences $\Gamma \cup \Delta$ and every sentence D in Δ is implied by the set of sentences Γ, then E is implied by Γ.

10.5 Let $\varnothing$ be the empty set of sentences, and let $\perp$ be any sentence that is not true on any interpretation. Show that:

(a) A sentence D is valid if and only if D is a consequence of $\varnothing$.

(b) A set of sentences Γ is unsatisfiable if and only if $\perp$ is a consequence of Γ.

10.6 Show that:
- **(a)** $\{C_1, \ldots, C_m\}$ is unsatisfiable if and only if $\sim C_1 \vee \cdots \vee \sim C_m$ is valid.
- **(b)** D is a consequence of $\{C_1, \ldots, C_m\}$ if and only if $\sim C_1 \vee \cdots \vee \sim C_m \vee D$ is valid.
- **(c)** D is a consequence of $\{C_1, \ldots, C_m\}$ if and only if $\{C_1, \ldots, C_m, \sim D\}$ is unsatisfiable.
- **(d)** D is valid if and only if $\sim D$ is unsatisfiable.

10.7 Show that $B(t)$ and $\exists x(x = t \,\&\, B(x))$ are logically equivalent.

10.8 Show that:
- **(a)** $(B \,\&\, C)$ implies B and implies C.
- **(b)** $\sim B$ implies $\sim(B \,\&\, C)$, and $\sim C$ implies $\sim(B \,\&\, C)$.
- **(c)** $\forall x\, B(x)$ implies $B(t)$.
- **(d)** $\sim B(t)$ implies $\sim \forall x\, B(x)$.

10.9 Show that:
- **(a)** If $\Gamma \cup \{\sim(B \,\&\, C)\}$ is satisfiable, then either $\Gamma \cup \{\sim B\}$ is satisfiable or $\Gamma \cup \{\sim C\}$ is satisfiable.
- **(b)** If $\Gamma \cup \{\sim \forall x\, B(x)\}$ is satisfiable, then for any constant c not occurring in Γ or $\forall x\, B(x)$, $\Gamma \cup \{\sim B(c)\}$ is satisfiable.

10.10 Show that the following hold for equivalence over any interpretation (and hence for logical equivalence), for any sentences (and hence for any formulas):
- **(a)** F is equivalent to F.
- **(b)** If F is equivalent to G, then G is equivalent to F.
- **(c)** If F is equivalent to G and G is equivalent to H, then F is equivalent to H.
- **(d)** If F and G are equivalent, then $\sim F$ and $\sim G$ are equivalent.
- **(e)** If F_1 and G_1 are equivalent, and F_2 and G_2 are equivalent, then $F_1 \,\&\, F_2$ and $G_1 \,\&\, G_2$ are equivalent, and similarly for $\vee$.
- **(f)** If c does not occur in $F(x)$ or $G(x)$, and $F(c)$ and $G(c)$ are equivalent, then $\forall x\, F(x)$ and $\forall x\, G(x)$ are equivalent, and similarly for $\exists$.

10.11 (*Substitution of equivalents*.) Show that the following hold for equivalence over any interpretation (and hence for logical equivalence):
- **(a)** If sentence G results from sentence F on replacing each occurrence of an atomic sentence A by an equivalent sentence B, then F and G are equivalent.
- **(b)** Show that the same holds for an atomic formula A and an equivalent formula B (provided, to avoid complications, that no variable occurring in A occurs bound in B or F).
- **(c)** Show that the same holds even when A is not atomic.

10.12 Show that $F(x)$ is (logically) equivalent to $G(x)$ if and only if $\forall x(F(x) \leftrightarrow G(x))$ is valid.

10.13 (*Relettering bound variables*.) Show that:
- **(a)** If F is a formula and y a variable not occurring free in F, then F is (logically) equivalent to a formula in which y does not occur at all. The same applies to any number of variables $y_1, \ldots, y_n$.
- **(b)** Every formula is logically equivalent to a formula having no subformulas in which the same variable occurs both free and bound.

10.14 Show that the following pairs are equivalent:

(a) $\forall x F(x) \,\&\, \forall y G(y)$ and $\forall u(F(u) \,\&\, G(u))$.

(b) $\forall x F(x) \vee \forall y G(y)$ and $\forall u \forall v(F(u) \vee G(v))$.

(c) $\exists x F(x) \,\&\, \exists y G(y)$ and $\exists u \exists v(F(u) \,\&\, G(v))$.

(d) $\exists x F(x) \vee \exists y G(y)$ and $\exists u(F(u) \vee G(u))$.

[In (a), it is to be understood that u may be a variable not occurring free in $\forall x F(x)$ or $\forall y G(y)$; in particular, if x and y are the same variable, u may be that same variable. In (b) it is to be understood that u and v may be any distinct variables not occurring free in $\forall x F(x) \vee \forall y G(y)$; in particular, if x does not occur in free in $\forall y G(y)$ and y does not occur free in $\forall x F(x)$, then u may be x and y may be v. Analogously for (d) and (c).]