Chapter 1

Classical Logic I: First-Order Logic

Wilfrid Hodges

1.1. First-Order Languages

The word 'logic' in the title of this chapter is ambiguous.

In its first meaning, a *logic* is a collection of closely related artificial languages. There are certain languages called *first-order languages*, and together they form first-order logic. In the same spirit, there are several closely related languages called modal languages, and together they form modal logic. Likewise second-order logic, deontic logic and so forth.

In its second but older meaning, *logic* is the study of the rules of sound argument. First-order languages can be used as a framework for studying rules of argument; logic done this way is called *first-order logic*. The contents of many undergraduate logic courses are first-order logic in this second sense.

This chapter will be about first-order logic in the first sense: a certain collection of artificial languages. In Hodges (1983), I gave a description of first-order languages that covers the ground of this chapter in more detail. That other chapter was meant to serve as an introduction to first-order logic, and so I started from arguments in English, gradually introducing the various features of first-order logic. This may be the best way in for beginners, but I doubt if it is the best approach for people seriously interested in the philosophy of first-order logic; by going gradually, one blurs the hard lines and softens the contrasts. So, in this chapter, I take the opposite route and go straight to the first-order sentences. Later chapters have more to say about the links with plain English.

The chief pioneers in the creation of first-order languages were Boole, Frege and C. S. Peirce in the nineteenth century; but the languages became public knowledge only quite recently, with the textbook of Hilbert and Ackermann (1950), first published in 1928 but based on lectures of Hilbert in 1917–22. (So first-order logic has been around for about 70 years, but Aristotle's syllogisms for well over 2000 years. Will first-order logic survive so long?)

From their beginnings, first-order languages have been used for the study of deductive arguments, but not only for this – both Hilbert and Russell used first-order formulas as an aid to definition and conceptual analysis. Today, computer science has still more uses for first-order languages, e.g., in knowledge representation and in specifying the behavior of systems.

You might expect at this point to start learning what various sentences in first-order languages *mean*. However, first-order sentences were never intended to mean anything; rather they were designed to express *conditions which things can satisfy or fail to satisfy*. They do this in two steps.

First, each first-order language has a number of symbols called *nonlogical constants*; older writers called them *primitives*. For brevity, I shall call them simply *constants*. To use a first-order sentence ϕ, something in the world – a person, a number, a colour, whatever – is attached (or in the usual jargon, *assigned*) to each of the constants of ϕ. There are some restrictions on what kind of thing can be assigned to what constant; more on that later. The notional glue that does the attaching is called an *interpretation* or a *structure* or a *valuation*. These three words have precise technical uses, but for the moment 'interpretation' is used as the least technical term.

Second, given a first-order sentence ϕ and an interpretation I of ϕ, the *semantics* of the first-order language determine either that I makes ϕ true, or that I makes ϕ false. If I makes ϕ true, this is expressed by saying that I *satisfies* ϕ, or that I is a *model* of ϕ, or that ϕ is *true in I* or *under I*. (The most natural English usage seems to be 'true in a structure' but 'true under an interpretation.' Nothing of any importance hangs on the difference between 'under' and 'in,' and I will not be entirely consistent with them.) The *truth-value* of a sentence under an interpretation is Truth if the interpretation makes it true, and Falsehood if the interpretation makes it false.

The main difference between one first-order language and any other lies in its set of constants; this set is called the *signature* of the language. (First-order languages can also differ in other points of notation, but this shall be ignored here.) If σ is a signature of some first-order language, then an interpretation is said to be *of signature* σ if it attaches objects to exactly those constants that are in σ. So an interpretation of signature σ contains exactly the kinds of assignment that are needed to make a sentence of signature σ true or false.

Examples of first-order languages must wait until some general notions are introduced in the next section, but as a foretaste, many first-order languages have a sentence that is a single symbol

$$\perp$$

pronounced 'absurdity' or 'bottom.' Nobody knows or cares what this symbol *means*, but the semantic rules decree that it is false. So, it has no models. It is not a nonlogical constant; its truth-value does not depend on any assignment.

1.2. Some Fundamental Notions

In the definitions below, it is assumed that some fixed signature σ has been chosen; the sentences are those of the first-order language of signature σ and the interpretations are of signature σ. So each interpretation makes each sentence either true or false:

$$\begin{array}{c}\text{true? false?}\\ \text{interpretations } I \leftrightarrow \text{sentences } \phi\end{array}$$

This picture can be looked at from either end. Starting from an interpretation I, it can be used as a knife to cut the class of sentences into two groups: the sentences which it satisfies and the sentences which it does not satisfy. The sentences satisfied by I are together known as the (first-order) *theory of I*. More generally, any set of sentences is called a *theory*, and I is a *model* of a theory T if it is a model of every sentence in T. By a standard mathematical convention, every interpretation is a model of the empty theory, because the empty theory contains no sentence that is false in the interpretation.

Alternatively, the picture can be read from right to left, starting with a sentence ϕ. The sentence ϕ separates the class of interpretations into two collections: those which satisfy it and those which do not. Those which satisfy ϕ are together known as the *model class* of ϕ. In fact, a similar definition can be given for any theory T: the *model class* of T is the class of all interpretations that satisfy T. If a particular class $\mathbf{K}$ of interpretations is the model class of a theory T, then T is a *set of axioms* for $\mathbf{K}$. This notion is important in mathematical applications of first-order logic, because many natural classes of structures – e.g., the class of groups – are the model classes of first-order axioms.

Two theories are said to be *logically equivalent*, or more briefly *equivalent*, if they have the same model class. As a special case, two sentences are said to be *equivalent* if they are true in exactly the same interpretations. A theory is said to be (semantically) *consistent* if it has at least one model; otherwise, it is (semantically) *inconsistent*. There are many semantically inconsistent theories, for example the one consisting of the single sentence '$\perp$'. The word 'semantically' is a warning of another kind of inconsistency, discussed at the end of section 1.8.

Suppose T is a theory and ψ is a sentence. Then T *entails* ψ if there is no interpretation that satisfies T but not ψ. Likewise, ψ is *valid* if every interpretation makes ψ true. One can think of validity as a special case of entailment: a sentence is valid if and only if it is entailed by the empty theory.

The symbol '$\vdash$' is pronounced 'turnstile.' A *sequent* is an expression

$$T \vdash \psi$$

where T on the left is a theory and ψ on the right is a sentence. The sentences in T are called the *premises* of the sequent and ψ is called its *conclusion*. The sequent is *valid* if T entails ψ, and *invalid* otherwise. If T is a finite set of sentences, the sequent '$T \vdash \psi$' can be written as a *finite sequent*

$$\phi_1, \ldots, \phi_n \vdash \psi$$

listing the contents of T on the left. The language under discussion (i.e. the first-order language of signature σ) is said to be *decidable* if there is an algorithm (i.e. a mechanical method which always works) for telling whether any given finite sequent is valid.

A *proof calculus* $\mathcal{C}$ consists of

(i) a set of rules for producing patterns of symbols called *formal proofs* or *derivations*, and
(ii) a rule which determines, given a formal proof and a sequent, whether the formal proof is a *proof* of the sequent.

Here 'proof of' is just a set of words; but one of the purposes of proof calculi is that they should give 'proofs of' all and only the valid sequents. The following definitions make this more precise:

1 A sequent

$$T \vdash \psi$$

is *derivable in* $\mathcal{C}$, or in symbols

$$T \vdash_{\mathcal{C}} \psi$$

if some formal proof in the calculus $\mathcal{C}$ is a proof of the sequent.
2 A proof calculus $\mathcal{C}$ is *correct* (or *sound*) if no invalid sequent is derivable in $\mathcal{C}$.
3 $\mathcal{C}$ is *complete* if every valid sequent is derivable in $\mathcal{C}$.

So a correct and complete proof calculus is one for which the derivable sequents are exactly the valid ones. One of the best features of first-order logic, from almost anybody's point of view, is that it has several excellent correct and complete proof calculi. Some are mentioned in section 1.8.

1.3. Grammar and Semantics

As in any language, the sentences of first-order languages have a grammatical structure. The details vary from language to language, but one feature that all first-order languages share is that *the grammatical structure of any given sentence is uniquely determined*. There are no grammatically ambiguous sentences like Chomsky's

They are flying planes.

This property of first-order languages is called the *unique parsing property*.

To guarantee unique parsing, first-order formulas generally have a large number of brackets. There are conventions for leaving out some of these brackets without introducing any ambiguity in the parsing. For example, if the first and last symbols of a sentence are brackets, they can be omitted. Any elementary textbook gives further details.

For historical reasons, there is a hitch in the terminology. With a first-order language, the objects that a linguist would call 'sentences' are called *formulas* (or in some older writers *well-formed formulas* or wff), and the word 'sentence' is reserved for a particular kind of formula, as follows.

Every first-order language has an infinite collection of symbols called *variables*:

$$x_0, x_1, x_2, \ldots$$

To avoid writing subscripts all the time, it is often assumed that

$$x, y, z, u, v$$

and a few similar symbols are variables too. Variables are not in the signature. From a semantic point of view, variables can occur in two ways: when a variable at some point in a formula needs to have an object assigned to it to give the formula a truth-value, this occurrence of the variable is called *free*; when no such assignment is needed, it is called *bound*. A *sentence* is a formula with no free occurrences of variables. To avoid confusing variables with constants, an assignment of objects to the variables is called a *valuation*. So, in general, a first-order formula needs an interpretation I of its constants *and* a valuation v of its variables to have a truth-value. (It will always be clear whether 'v' means a variable or a valuation.)

The definitions of the previous section all make sense if 'sentence' is read as 'first-order sentence'; they also make sense if 'sentence' is read as 'first-order formula' and 'interpretation' as 'interpretation plus valuation'. Fortunately, the two readings do not clash; for example, a sequent of first-order sentences is valid or invalid, regardless of whether the first-order sentences are regarded as sentences or as formulas. That needs a proof – one that can be left to the mathematicians. Likewise, according to the mathematicians, a first-order language is decidable in terms of sentences if and only if it is decidable in terms of formulas. (Be warned though that 'first-order theory' normally means 'set of first-order sentences' in the narrow sense. To refer to a set of first-order formulas it is safest to say 'set of first-order formulas.')

The next few sections present the semantic rules in what is commonly known as the *Tarski style* (in view of Tarski (1983) and Tarski and Vaught (1957). In this style, to find out what interpretations-plus-valuations make a complex formula ϕ true, the question is reduced to the same question for certain formulas that are simpler than ϕ. The Tarski style is not the only way to present the semantics. A suggestive alternative is the Henkin–Hintikka description in terms of games; see Hintikka (1973, ch. V), or its computer implementation by Barwise and Etchemendy (1999). Although the Tarski-style semantics and the game semantics look very different, they always give the same answer to the question: 'What is the truth-value of the sentence ϕ under the interpretation I?'

Throughout this chapter, symbols such as 'ϕ', 'α' are used to range over the formulas or terms of a first-order language. They are *metavariables*; in other words, they are not in the first-order languages but are used for talking about these languages. On the other hand, so as not to saddle the reader with still more metavariables for the other expressions of a first-order language, for example, when making a general statement about variables, a typical variable may be used as if it was a meta-variable. Thus 'Consider a variable x' is common practice. More generally, quotation marks are dropped when they are more of a nuisance than a help.

1.4. The First-Order Language with Empty Signature

The simplest first-order language is the one whose signature is empty. In this section, this language is referred to as L.

An interpretation I with empty signature does not interpret any constants, but – for reasons that will appear very soon – it does have an associated class of objects, called its *universe* or *domain*. (The name 'domain' is perhaps more usual; but it has other meanings in logic so, to avoid confusion, 'universe' is used instead.) Most logicians require that the universe shall have at least one object in it; but, apart from this, it can be any class of objects. The members of the domain are called the *elements* of the interpretation; some older writers call them the *individuals*. A *valuation in* I is a rule v for assigning to each variable x_i an element $v(x_i)$ in the universe of I.

For the grammatical constructions given here, an interpretation I and a valuation v in I are assumed. The truth-values of formulas will depend partly on I and partly on v. A formula is said to be *true in I under v*.

Some expressions of L are called *atomic formulas*. There are two kinds:

- Every expression of the form '$(x = y)$', where x and y are variables, is an atomic formula of L.
- '$\perp$' is an atomic formula of L.

It has already been noted that '$\perp$' is false in I. The truth-value of $(x_1 = x_3)$, to take a typical example of the other sort of atomic formula, is

Truth	if $v(x_1)$ is the same element as $v(x_3)$
Falsehood	if not

So, given I and v, truth-values are assigned to the atomic formulas.

Next, a class of expressions called the *formulas* of L is defined. The atomic formulas are formulas, but many formulas of L are not atomic. Take the five symbols

$$\sim \qquad \& \qquad \vee \qquad \supset \qquad \equiv$$

which are used to build up complex formulas from simpler ones. There is a grammatical rule for all of them.

If ϕ and ψ are formulas, then so are each of these expressions:

$$(\sim \phi) \qquad (\phi \,\&\, \psi) \qquad (\phi \vee \psi) \qquad (\phi \supset \psi) \qquad (\phi \equiv \psi)$$

This chart, called a *truth-table*, shows which of these formulas are true, depending on whether ϕ and ψ are true:

ϕ	ψ	$(\sim \phi)$	$(\phi \,\&\, \psi)$	$(\phi \vee \psi)$	$(\phi \supset \psi)$	$(\phi \equiv \psi)$
T	T	F	T	T	T	T
T	F		F	T	F	F
F	T	T	F	T	T	F
F	F		F	F	T	T

(Here T = Truth and F = Falsehood.) Because of this table, the five symbols '$\sim$', '&', '$\vee$', '$\supset$', and '$\equiv$' are known as the *truth-functional symbols.*

For example, the formula

$$(((x_2 = x_3) \,\&\, (x_5 = x_2)) \supset (x_5 = x_3))$$

is false in just one case, namely where $v(x_2)$ and $v(x_3)$ are the same element, and $v(x_5)$ and $v(x_2)$ are the same element, but $v(x_5)$ and $v(x_3)$ are not the same element. Since this case can never arise, the formula is true regardless of what I and v are.

There remain just two grammatical constructions. The grammatical rule for them both is:

> If ϕ is any formula and x is any variable, then the expressions
>
> $$(\forall x)\phi \qquad (\exists x)\phi$$
>
> are both formulas.

The expressions '$(\forall x)$' and '$(\exists x)$' are called respectively a *universal quantifier* and an *existential quantifier*, and read respectively as 'for all x' and 'there is x'. In the two formulas given by the rule, the occurrence of x inside the quantifier is said to *bind* itself and any occurrences of the *same* variable in the formula ϕ. These occurrences stay bound as still more complex formulas are built. In any formula, an occurrence of a variable which is not bound by a quantifier in the formula is said to be *free*. A formula with no free variables is called a *sentence.* (And this is what was meant earlier by 'sentences' of first-order languages. The syntactic definitions just given are equivalent to the semantic explanation in the previous section.) For example, this is not a sentence:

$$(\exists y)(\sim (x = y))$$

because it has a free occurrence of x (though both occurrences of y are bound by the existential quantifier). But this is a sentence:

$$(\forall x)(\exists y)(\sim (x = y))$$

because its universal quantifier binds the variable x.

The semantic rules for the quantifiers are one of the harder concepts of first-order logic. For more than two millennia, some of the best minds of Europe struggled to formulate semantic rules that capture the essence of the natural language expressions ‘all’ and ‘there is.’

- ‘$(\forall x)\phi$’ is true in I under v if: for every element a in the universe of I, if w is taken to be the valuation exactly like v except that $w(x)$ is a, then ϕ is true in I under w.
- ‘$(\exists x)\phi$’ is true in I under v if: there is an element a in the universe of I, such that if w is taken to be the valuation exactly like v except that $w(x)$ is a, then ϕ is true in I under w.

For example, the formula

$$(\exists y)(\sim (x = y))$$

is true in I under v, iff (if and only if) there is an element a such that $v(x)$ is not the same element as a. So the sentence

$$(\forall x)(\exists y)(\sim (x = y))$$

is true in I under v iff for every element b there is an element a such that b is not the same element as a. In other words, it is true iff the universe of I contains at least two different elements.

Note that this last condition depends only on I and not at all on v. One can prove that the truth-value of a formula ϕ in I and v never depends on $v(x)$ for any variable x that does not occur free in ϕ. Since sentences have no free variables, their truth-value depends only on I and the valuation slips silently away.

These rules capture the essence of the expressions ‘all’ and ‘there is’ by stating precise conditions under which a sentence starting with one of these phrases counts as true. The same applies to the truth-functional symbols, which are meant, in some sense, to capture at least the mathematical use of the words ‘not’, ‘and’, ‘or’, ‘if . . . then’, and ‘if and only if’.

1.5. Some Notation

The notation in this section applies to all first-order languages, not just the language with empty signature.

Writing a formula as $\phi(x_1, \ldots, x_n)$, where $x_1, \ldots, x_n$ are different variables means that the formula is ϕ and it has no occurrences of free variables except perhaps for $x_1, \ldots, x_n$. Then

$$I \models \phi[a_1, \ldots, a_n]$$

means that ϕ is true in the interpretation I and under some valuation v for which $v(x_1), \ldots, v(x_n)$ are $a_1, \ldots, a_n$ respectively (or under any such valuation v – it makes no difference). When ϕ is a sentence, the $a_1, \ldots, a_n$ are redundant and

$$I \vDash \phi$$

simply means that ϕ is true in I.

Here is another useful piece of notation:

$$\phi(y/x)$$

means the formula obtained by replacing each free occurrence of x in ϕ by an occurrence of y. Actually, this is not quite right, but the correction in the next paragraph is rather technical. What is intended is that the formula $\phi(y/x)$ 'says about y' the same thing that ϕ 'said about x.'

Suppose, for example, that ϕ is the formula

$$(\forall y)(x = y)$$

which expresses that x is identical with everything. Simply putting y in place of each free occurrence of x in ϕ, gives

$$(\forall y)(y = y)$$

This says that each thing is identical to itself; whereas the intention was to make the more interesting statement that y is identical with everything. The problem is that y is put into a place where it immediately becomes bound by the quantifier $(\forall y)$. So $\phi(y/x)$ must be defined more carefully, as follows. First, choose another variable, say z, that does not occur in ϕ, and adjust ϕ by replacing all bound occurrences of y in ϕ by bound occurrences of z. After this, substitute y for free occurrences of x. (So $\phi(y/x)$ in our example now works out as

$$(\forall z)(y = z)$$

which says the right thing.) This more careful method of substitution is called *substitution avoiding clash of variables.*

The language L of the previous section is a very arid first-order language. The conditions that it can express on an interpretation I are very few. It can be used to say that I has at least one element, at least two elements, at least seven elements, either exactly a hundred or at least three billion elements, and similar things; but nothing else. (Is there a single sentence of L which expresses the condition that I has infinitely many elements? No. This is a consequence of the compactness theorem in section 1.10.)

Nevertheless L already shows some very characteristic features of first-order languages. For example, to work out the truth-value of a sentence ϕ under an interpretation I, one must generally consider the truth-values of subformulas of ϕ

under various valuations. As explained in section 1.3, the notion of a valid sequent applies to formulas as well as sentences; but for formulas it means that every interpretation-plus-valuation making the formulas on the left true makes the formula on the right true too.

Here are two important examples of valid sequents of the language *L*. The sequent

$$\vdash (x = x)$$

is valid because $v(x)$ is always the same element as $v(x)$. The sequent

$$(x = y) \vdash (\phi \supset \phi(y/x))$$

is valid because if two given elements are the same element, then they satisfy all the same conditions.

1.6. Nonlogical Constants: Monadic First-Order Logic

Section 1.5 ignored the main organ by which first-order formulas reach out to the world: the signature, the family of nonlogical constants.

The various constants can be classified by the kinds of feature to which they have to be attached in the world. For example, some constants are called *class symbols* because their job is to stand for classes. (Their more technical name is 1-*ary relation symbols.*) Some other constants are called *individual constants* because their job is to stand for individuals, i.e. elements. This section concentrates on languages whose signature contains only constants of these two kinds. Languages of this type are said to be *monadic*. Let *L* be a monadic language.

Usually, individual constants are lower-case letters '*a*', '*b*', '*c*' etc. from the first half of the alphabet, with or without subscripts. Usually class symbols are capital letters '*P*', '*Q*', '*R*' etc. from the second half of the alphabet, with or without number subscripts.

Grammatically these constants provide some new kinds of atomic formula. It is helpful first to define the *terms* of *L*. There are two kinds:

- Every variable is a term.
- Every individual constant is a term.

The definition of *atomic formula* needs revising:

- Every expression of the form '$(\alpha = \beta)$', where α and β are terms, is an atomic formula of *L*.
- If *P* is any class symbol and α any term, then '$P(\alpha)$' is an atomic formula.
- '$\perp$' is an atomic formula of *L*.

Apart from these new clauses, the grammatical rules remain the same as in section 1.4.

What should count as an interpretation for a monadic language? Every interpretation I needs a universe, just as before. But now it also needs to give the truth-value of $P(x)$ under a valuation that ties x to an element $v(x)$, which might be any element of the universe. In other words, the interpretation needs to give the class, written P^I, of all those elements a such that $P(x)$ is true under any valuation v with $v(x)$ equal to a. (Intuitively P^I is the class of all elements that satisfy $P(x)$ in I.) Here P^I might be any subclass of the universe.

In the branch of logic called model theory, the previous paragraph turns into a definition. A *structure* is an interpretation I of the following form:

- I has a universe, which is a set (generally taken to be non-empty).
- For each class symbol P in the signature, I picks out a corresponding class P^I, called the *interpretation of P under I*, all of whose members are in the universe.
- For each individual constant a in the signature, I picks out a corresponding element a^I in the universe, and this element is called the *interpretation of a under I.*

Writing σ for the signature in question, this interpretation I is called a *σ-structure.* So a σ-structure contains exactly the information needed to give a truth-value to a sentence of signature σ. Note that the interpretations a^I are needed to deal with sentences such as $P(a)$. (The requirement that the universe should be a set rather than a class is no accident: a set is a mathematically well-behaved class. The precise difference is studied in texts of set theory. [See chapter 3.])

However, this model-theoretic definition is not as old as first-order logic. In the early days, logicians would give an interpretation for a by writing down a name or a description of a thing or person. They would give an interpretation for P by writing down a sentence of English or their own native language with x in place of one or more pronouns. Or sometimes they would write a sentence with 'He' or 'It' left off the front; or more drastically, a sentence with 'He is a' left off. For example an interpretation might contain the items:

P : x is kind to people who are kind to x.

Q : is mortal

R : taxpayer

The third style here is the least flexible, but anyway it is not needed; it can easily be converted to the second style by writing 'is a taxpayer.' The second style in turn is less flexible than the first, and again is not needed. Q and R could be written 'x is mortal', 'x is a taxpayer'. A sentence with variables in place of some pronouns is sometimes called a *predicate.*

Can every interpretation by predicates be converted into a structure? Yes, provided that each of the predicates has a certain property: *the question whether an element of the universe satisfies the predicate always has a definite answer* (*Yes or No*) *which depends only on the element and not on how it is described.* Predicates with this property are said to be *extensional.* The following predicates seem not to be extensional

– though this is an area where people have presented strong arguments for some quite surprising conclusions:

> x is necessarily equal to 7.
>
> I recognized x.

The prevailing view is that to handle predicates like these, a logic with a subtler semantics than first-order logic is needed. Modal logic takes on board the first example, epistemic logic the second. [See chapters 7 and 9.]

The predicate

> x is bald.

also fails the test, not because it is possible to be bald under one name and bushy-haired under another, but because there are borderline cases – people who aren't definitely bald or definitely not bald. So this predicate does not succeed in defining a class of people. Truth to tell, most natural language predicates are at least slightly vague; even logicians have to live with the roughnesses of the real world.

Given an interpretation I that uses predicates, a first-order sentence ϕ can often be translated into an English sentence which is guaranteed to be true if and only if ϕ is true in I. The translation will generally need to mention the universe of the interpretation, unless a predicate is used to describe that too. Here are some examples, using the interpretation a couple of paragraphs above, together with the universe described by 'x is a person':

> $(\forall x)(R(x) \to Q(x))$
> Every person who is a taxpayer is mortal.
>
> $(\exists x)P(x)$
> At least one person is kind to people who are kind to him or her.
>
> $(\exists x)(R(x) \,\&\, (\forall y)(R(y) \equiv (y = x)))$
> Exactly one person is a taxpayer.

The reader may well agree with the following comment: If these first-order sentences are being used to express the English sentences in question, then it is artificial to ask for a universe at all. In ordinary speech, no one asks people to state their universes.

This comment needs answers on several levels. First, mathematical objects – such as groups, rings, boolean algebras and the like – consist of a set of objects with certain features picked out by nonlogical constants. So it was natural for the mathematical creators of first-order logic to think of this set of objects as a universe.

Second, there is a mathematical result that takes some of the sting out of the requirement that a universe has to be chosen. An occurrence of a universal quantifier is *restricted* if it occurs as follows:

$(\forall x)(P(x) \supset \cdots)$

i.e. followed immediately by a left bracket, a class symbol with the same variable, and then '$\supset$'. Likewise an occurrence of an existential quantifier is *restricted* if it looks like this:

$(\exists x)(P(x) \,\&\, \cdots)$

The mathematical result states:

Theorem 1.1 Let ϕ be a sentence of the first-order language L of signature σ, and suppose that all occurrences of quantifiers in ϕ are restricted. Let I be a σ-structure and J be another σ-structure which comes from I by removing some elements which are not inside P^I for any class symbol P. Then ϕ has the same truth-value in I and in J.

First-order sentences that serve as straightforward translations of English sentences usually have all their quantifiers restricted, as in the first and third examples above. (The second example can be rewritten harmlessly as

$(\exists x)(P(x) \,\&\, P(x))$

and then its quantifier is restricted too.) So the choice of universe may be largely *ad hoc*, but it is also largely irrelevant. (This theorem remains true for first-order languages that are not monadic.)

Third, if the class symbols are interpreted by predicates rather than by classes, the choice of universe certainly can make a difference to truth-values, even for sentences covered by the theorem just stated. Suppose, for example, that an interpretation is being used, with

P : x is a person.

Q : x will be dead before the year 2200.

With such an interpretation, the sentence

$(\forall x)(P(x) \to Q(x))$

expresses that every person will be dead before the year 2200. This is probably true of people alive now, but probably false if 'person' includes people yet to be born. So different universes give different truth-values. Why does this not contradict Theorem 1.1? Because the predicate 'x is a person' picks out different classes according as future people are excluded or included, so that the corresponding σ-structures differ in their assignments to P and Q, not just in their universes.

If a universe can contain future people, can it contain possible people, or fictional people, or even impossible people (like the man I met who wasn't there, in the children's jingle)? Or to be more metaphysical, can a universe contain as separate elements myself-now and myself-ten-years-ago? First-order logic is very robust about questions like these: it doesn't give a damn. If you think that there are fictional people and that they have or fail to have this property or that, and can meaningfully be said to be the same individuals or not the same individuals as one another, then fine, put them in your universes. Likewise, if you think there are time-slices of people. If you don't, then leave them out.

All these remarks about universes apply equally well to the more general first-order languages of section 1.7. Here is a theorem that does not.

Theorem 1.2 If L is a monadic first-order language with a finite signature, then L is decidable.

See, for example, Boolos and Jeffrey (1974, ch. 25, 'Monadic versus dyadic logic') for a proof of this theorem.

1.7. Some More Nonlogical Constants

Most logicians before about 1850, if they had been set to work designing a first-order language, would probably have been happy to stick with the kinds of constant already introduced here. Apart from some subtleties and confusions about empty classes, the traditional syllogistic forms correspond to the four sentence-types

$$(\forall x)(P(x) \supset Q(x)) \qquad (\forall x)(P(x) \supset \sim Q(x))$$
$$(\exists x)(P(x) \,\&\, Q(x)) \qquad (\exists x)(P(x) \,\&\, \sim Q(x))$$

The main pressure for more elaborate forms came from mathematics, where geometers wanted symbols to represent predicates such as:

x is a point lying on the line y.

x is between y and z.

Even these two examples show that there is no point in restricting ourselves in advance to some fixed number of variables. So, class symbols are generalized to *n-ary relation symbols*, where the *arity*, n, is the number of distinct variables needed in a predicate that interprets the relation symbol.

Like class symbols, relation symbols are usually 'P', 'Q', etc., i.e. capital letters from near the end of the alphabet. An *ordered n-tuple* is a list

$$\langle a_1, \ldots, a_n \rangle$$

where a_i is the ith item in the list; the same object may appear as more than one item in the list. The interpretation R^I of a relation symbol R of arity n in an interpretation I is a set of ordered n-tuples of elements in the universe of I. If R^I is specified by giving a particular predicate for R, then which variables of the predicate belong with which places in the lists must also be specified. An example shows how:

$R(x, y, w)$: w is between x and y.

Class symbols are included as the relation symbols of arity 1, by taking a list

$\langle a \rangle$

of length 1 to be the same thing as its unique item a.

There can also be relation symbols of arity 0 if it is decided that there is exactly one list $\langle\, \rangle$ of length 0. So the interpretation p^I of a 0-ary relation symbol p is either the empty set (call it Falsehood) or else the set whose one element is $\langle\, \rangle$ (call this set Truth). All this makes good sense set-theoretically. What matters here, however, is the outcome: relation symbols of arity 0 are called *propositional symbols*, and they are always interpreted as Truth or as Falsehood. A sentence which contains neither '=', quantifiers nor any nonlogical constants except propositional symbols is called a *propositional sentence*. Propositional logic is about propositional sentences.

The language can be extended in another way by introducing nonlogical symbols called *n-ary function symbols*, where n is a positive integer. The interpretation F^I of such a symbol F is a function which assigns an element of I to each ordered n-tuple of elements of I. (Again, there is a way of regarding individual constants as 0-ary function symbols, but the details can be skipped here.)

The new symbols require some more adjustments to the grammar. The clause for *terms* becomes:

- Every variable is a term.
- Every individual constant is a term.
- If F is a function symbol of arity n, and $\alpha_1, \ldots, \alpha_n$ are terms, then '$F(\alpha_1, \ldots, \alpha_n)$' is a term.

The definition of *atomic formula* becomes:

- Every expression of the form '$(\alpha = \beta)$', where α and β are terms, is an atomic formula of L.
- Every propositional symbol is an atomic formula.
- If R is any relation symbol of positive arity n and $\alpha_1, \ldots, \alpha_n$ are terms, then '$R(\alpha_1, \ldots, \alpha_n)$' is an atomic formula.
- '$\bot$' is an atomic formula.

The semantic rules are the obvious adjustments of those in the previous section.

Some notation from section 1.5 can be usefully extended. If ϕ is a formula and α is a term,

$\phi(\alpha/x)$

represents the formula obtained from ϕ by replacing all free occurrences of x by α. As in section 1.5, to avoid clash of variables, the bound variables in ϕ may need to be changed first, so that they do not bind any variables in α.

1.8. Proof Calculi

First-order logic has a range of proof calculi. With a very few exceptions, all these proof calculi apply to all first-order languages. So, for the rest of this section assume that L is a particular first-order language of signature σ.

The first proof calculi to be discovered were the *Hilbert-style* calculi, where one reaches a conclusion by applying deduction rules to axioms. An example is described later in this section. These calculi tend to be very *ad hoc* in their axioms, and maddeningly wayward if one is looking for proofs in them. However, they have their supporters, e.g., modal logicians who need a first-order base to which further axioms can be added.

In 1934, Gentzen (1969) invented two other styles of calculus. One was the *natural deduction calculus* (independently proposed by Jaśkowski slightly earlier). An intuitionistic natural deduction calculus is given in chapter 11, which, as noted there, can be extended to make a calculus for classical first-order logic by the addition of a rule for double-negation elimination. Gentzen's second invention was the *sequent calculus*, which could be regarded as a Hilbert-style calculus for deriving finite sequents instead of formulas. With this subtle adjustment, nearly all of the arbitrariness of Hilbert-style systems falls away, and it is even possible to convert each sequent calculus proof into a sequent calculus proof in a very simple form called a *cut-free proof*. The popular *tableau* or *truth-tree* proofs are really cut-free sequent proofs turned upside down. A proof of a sequent in any of the four kinds of calculi – Hilbert-style, natural deduction, sequent calculus, tableaux – can be mechanically converted to a proof of the same sequent in any of the other calculi; see Sundholm (1983) for a survey.

The *resolution calculus* also deserves a mention. This calculus works very fast on computers, but its proofs are almost impossible for a normal human being to make any sense of, and it requires the sentences to be converted to a normal form (not quite the one in section 1.10 below) before the calculation starts; see, for example, Gallier (1986).

To sketch a Hilbert-style calculus, called $\mathcal{H}$, first define the class of *axioms* of $\mathcal{H}$. This is the set of all formulas of the language L which have any of the following forms:

H1 $\phi \supset (\psi \supset \phi)$

H2 $(\phi \supset \psi) \supset ((\phi \supset (\psi \supset \chi)) \supset (\psi \supset \chi))$

H3 $(\sim \phi \supset \psi) \supset ((\sim \phi \supset \sim \psi) \supset \phi)$

H4 $((\phi \supset \bot) \supset \bot) \supset \phi$

H5 $\phi \supset (\psi \supset (\phi \,\&\, \psi))$

H6 $(\phi \,\&\, \psi) \supset \phi$, $(\phi \,\&\, \psi) \supset \psi$

H7 $\phi \supset (\phi \vee \psi)$, $\psi \supset (\phi \vee \psi)$

H8 $(\phi \supset \chi) \supset ((\psi \supset \chi) \supset ((\phi \vee \psi) \supset \chi))$

H9 $(\phi \supset \psi) \supset ((\psi \supset \phi) \supset (\phi \equiv \psi))$

H10 $(\phi \equiv \psi) \supset (\phi \supset \psi)$, $(\phi \equiv \psi) \supset (\psi \supset \phi)$

H11 $\phi(\alpha/x) \supset \exists x\phi$ (α any term)

H12 $\forall x\phi \supset \phi(\alpha/x)$ (α any term)

H13 $x = x$

H14 $x = y \supset (\phi \supset \phi(y/x))$

A *derivation* (or *formal proof*) in $\mathcal{H}$ is defined to be a finite sequence

$$(\langle \phi_1, m_1 \rangle, \ldots, \langle \phi_n, m_n \rangle)$$

such that $n \geq 1$, and for each i ($1 \leq i \leq n$) one of the five following conditions holds. (Clauses (c)–(e) are known as the *derivation rules* of $\mathcal{H}$.)

(a) $m_i = 1$ and ϕ_i is an axiom.
(b) $m_i = 2$ and ϕ_i is any formula of L.
(c) $m_i = 3$ and there are j and k in $\{1, \ldots, i-1\}$ such that ϕ_k is $\phi_j \to \phi_i$.
(d) $m_i = 4$ and there is j ($1 \leq j < i$) such that ϕ_j has the form $\psi \to \chi$, x is a variable not occurring free in ψ, and ϕ_i is $\psi \to \forall x\chi$.
(e) $m_i = 5$ and there is j ($1 \leq j < i$) such that ϕ_j has the form $\psi \to \chi$, x is a variable not occurring free in χ, and ϕ_i is $\exists x\psi \to \chi$.

The *premises* of this derivation are the formulas ϕ_i for which $m_i = 2$. Its *conclusion* is ϕ_n. We say that ψ is *derivable in $\mathcal{H}$* from a set T of formulas, in symbols

$$T \vdash_{\mathcal{H}} \psi$$

if there exists a derivation whose conclusion is ψ and all of whose premises are in T.

Proofs are usually written vertically rather than horizontally. For example here is a proof of $(\phi \supset \phi)$, where ϕ is any formula:

(1)	$(\phi \supset (\phi \supset \phi)) \supset ((\phi \supset ((\phi \supset \phi) \supset \phi)) \supset (\phi \supset \phi))$	[Axiom H2]
(2)	$\phi \supset (\phi \supset \phi)$	[Axiom H1]
(3)	$(\phi \supset ((\phi \supset \phi) \supset \phi)) \supset (\phi \supset \phi)$	[Rule (c) from (1), (2)]
(4)	$\phi \supset ((\phi \supset \phi) \supset \phi)$	[Axiom H1]
(5)	$\phi \supset \phi$	[Rule (c) from (3), (4)]

To save the labor of writing this argument every time a result of the form $\phi \supset \phi$ is needed, this result can be quoted as a lemma in further proofs. Thus $\sim\perp$ can be proved as follows:

(1) $\sim\perp\supset\sim\perp$ [Lemma]

(2) $(\sim\perp\supset\sim\perp) \supset ((\sim\perp\supset\perp) \supset (\sim\perp\supset\sim\perp))$ [Axiom H1]

(3) $(\sim\perp\supset\perp) \supset (\sim\perp\supset\sim\perp)$ [Rule (c) from (1), (2)]

(4) $((\sim\perp\supset\perp) \supset (\sim\perp\supset\sim\perp)) \supset (((\sim\perp\supset\perp) \supset ((\sim\perp\supset\sim\perp) \supset\perp)) \supset ((\sim\perp\supset\perp) \supset\perp))$ [Axiom H2]

(5) $((\sim\perp\supset\perp) \supset ((\sim\perp\supset\sim\perp) \supset\perp)) \supset ((\sim\perp\supset\perp) \supset\perp)$ [Rule (c) from (3), (4)]

(6) $(\sim\perp\supset\perp) \supset ((\sim\perp\supset\sim\perp) \supset\perp)$ [Axiom H3]

(7) $(\sim\perp\supset\perp) \supset\perp$ [Rule (c) from (5), (6)]

(8) $((\sim\perp\supset\perp) \supset\perp) \supset\sim\perp$ [Axiom H4]

(9) $\sim\perp$ [Rule (c) from (7), (8)]

Then this result can be quoted in turn as a lemma in a proof of $(\phi \supset \perp) \supset\sim \phi$, and so on.

A theory T is *$\mathcal{H}$-inconsistent* if there is some formula ϕ such that $(\phi \,\&\, \sim \phi)$ is derivable from T in $\mathcal{H}$. If the language L contains $\perp$, then it can be shown that this is equivalent to saying that $\perp$ is derivable from T in $\mathcal{H}$. T is *$\mathcal{H}$-consistent* if it is not $\mathcal{H}$-inconsistent. $\mathcal{H}$-inconsistency is one example of *syntactic inconsistency*; other proof calculi give other examples.

1.9. Correctness and Completeness

Theorem 1.3 (*Correctness Theorem for $\mathcal{H}$*) Suppose $\phi_1, \ldots, \phi_n$ and ψ are sentences. If ϕ is derivable in $\mathcal{H}$ from $\phi_1, \ldots, \phi_n$, then the sequent

$$\phi_1, \ldots, \phi_n \vdash \psi$$

is valid.

Proof sketch This is proved by induction on the length of the shortest derivation of ψ from $\phi_1, \ldots, \phi_n$. Unfortunately, the formulas in the derivation need not be sentences. So for the induction hypothesis something a little more general needs to be proved:

Suppose $\phi_1, \ldots, \phi_n$ are sentences and ψ is a formula whose free variables are all among $x_1, \ldots, x_m$. If ψ is derivable in $\mathcal{H}$ from $\phi_1, \ldots, \phi_n$, then the sequent

$$\phi_1, \ldots, \phi_n \vdash \forall x_1 \cdots \forall x_m \psi$$

is valid.

The argument splits into cases according to the last derivation rule used in the proof. Suppose, for example, that this was the rule numbered (5) above, and ψ is the formula $\exists y\theta \supset \chi$ where y is not free in χ. Then, from the induction hypothesis, the sequent

$$\phi_1, \ldots, \phi_n \vdash \forall x_1 \cdots \forall x_n \forall y(\theta \supset \chi)$$

is valid. Using the fact that y is not free in χ, it can be checked that the sequent

$$\forall x_1 \cdots \forall x_n \forall y(\theta \supset \chi) \vdash \forall x_1 \cdots \forall x_n(\exists y\theta \supset \chi)$$

is valid. By this and the induction hypothesis, the sequent

$$\phi_1, \ldots, \phi_n \vdash \forall x_1 \cdots \forall x_n(\exists y\theta \supset \chi)$$

is valid as required. QED

Now, the completeness question:

Theorem 1.4 (*Completeness Theorem for $\mathcal{H}$*) Suppose that T is a theory and ψ is a sentence such that the sequent

$$T \vdash \psi$$

is valid. Then ψ is derivable from T in $\mathcal{H}$.

In fact one proves the special case of the Completeness Theorem where ψ is $\perp$; in other words

If T is a theory with no models, then $T \vdash_{\mathcal{H}} \perp$.

This is as good as proving the whole theorem, since the sequent

$$T \cup \{\sim \psi\} \vdash \perp$$

is equivalent to '$T \vdash \psi$' both semantically and in terms of derivability in $\mathcal{H}$.

Here, the Completeness Theorem is proved by showing that if T is any $\mathcal{H}$-consistent theory then T has a model. A technical lemma about $\mathcal{H}$ is needed along the way:

Lemma 1.5 Suppose c is a constant which occurs nowhere in the formula ϕ, the theory T or the sentence ψ. If

$$T \vdash_{\mathcal{H}} \phi(c/x) \supset \psi$$

then

$$T \vdash_{\mathcal{H}} \exists x\phi \supset \psi$$

Proof sketch of the Completeness Theorem This is known as a *Henkin-style* proof because of three features: the constants added as witnesses, the construction of a maximal consistent theory, and the way that a model is built using sets of terms as elements. The proof uses a small amount of set theory, chiefly infinite cardinals and ordinals. [See chapter 3.]

Assume a $\mathcal{H}$-consistent theory T in the language L. Let κ be the number of formulas of L; κ is always infinite. Expand the language L to a first-order language L^+ by adding to the signature a set of κ new individual constants; these new constants are called *witnesses.* List the sentences of L^+ as $(\phi_i : i < \kappa)$. Now define for each $i < \kappa$ a theory T_i, so that

$$T = T_0 \subseteq T_1 \subseteq \cdots$$

and each T_i is $\mathcal{H}$-consistent. To start the process, put $T_0 = T$. When i is a limit ordinal, take T_i to be the union of the T_j with $j < i$; this theory is $\mathcal{H}$-consistent since any inconsistency would have a proof using finitely many sentences, all of which would lie in some T_j with $j < i$.

The important choice is where i is a successor ordinal, say $i = j + 1$. If $T_j \cup \{\phi_j\}$ is not $\mathcal{H}$-consistent, take T_{j+1} to be T_j. Otherwise, put $T'_j = T_j \cup \{\phi_j\}$. Then if ϕ_j is of the form $\exists x\psi$, choose a witness c that appears nowhere in any sentence of T'_j, and put $T_{j+1} = T'_j \cup \{\psi(c/x)\}$; otherwise put $T_{j+1} = T'_j$. By Lemma 1.5, T_{j+1} is $\mathcal{H}$-consistent in all these cases.

Write T^+ for the union of all the theories T_i. It has the property that if ϕ_j is any sentence of L^+ for which $T^+ \cup \{\phi_j\}$ is $\mathcal{H}$-consistent, then $T_j \cup \{\phi_j\}$ was already $\mathcal{H}$-consistent and so ϕ_j is in T^+ by construction. (As noted, T^+ is *maximal consistent.*) Moreover if T^+ contains a sentence ϕ_j of the form $\exists x\psi$, then by construction it also contains $\psi(c/x)$ for some witness c.

Two witnesses c and d are *equivalent* if the sentence '$c = d$' is in T^+. Now if '$c = d$' and '$d = e$' are both in T^+, then (appealing to the axioms and rules of $\mathcal{H}$) the theory $T^+ \cup \{c = e\}$ is $\mathcal{H}$-consistent, and so '$c = e$' is also in T^+. This and similar arguments show that 'equivalence' is an equivalence relation on the set of witnesses. Now build a structure A^+ whose universe is the set of equivalence classes $c^\sim$ of witnesses c. For example, if P is a 2-ary relation symbol in the signature, then take P^{A+} to be the set of all ordered pairs $\langle c^\sim, d^\sim\rangle$ such that the sentence '$P(c, d)$' is in T^+. There are a number of details to be checked, but the outcome is that A^+ is a model of T^+. Now, stripping the witnesses out of the signature gives a structure A whose signature is that of L, and A is a model of all the sentences of L that are in T^+. In particular, A is a model of T, as required. (Note that A has at most κ elements, since there were only κ witnesses.) QED

1.10. Metatheory of First-Order Logic

The *metatheory* of a logic consists of those things that one can say *about* the logic, rather than in it. All the numbered theorems of this chapter are examples. The metatheory of first-order logic is vast. Here are a few high points, beginning with some consequences of the Completeness Theorem for $\mathcal{H}$.

Theorem 1.6 (*Compactness Theorem*) Suppose T is a first-order theory, ψ is a first-order sentence and T entails ψ. Then there is a finite subset U of T such that U entails ψ.

Proof If T entails ψ then the sequent

$$T \vdash \psi$$

is valid, and so by the completeness of the proof calculus $\mathcal{H}$, the sequent has a formal proof. Let U be the set of sentences in T which are used in this proof. Since the proof is a finite object, U is a finite set. But the proof is also a proof of the sequent

$$U \vdash \psi$$

So by the correctness of $\mathcal{H}$, U entails ψ. QED

Corollary 1.7 Suppose T is a first-order theory and every finite subset of T has a model. Then T has a model.

Proof Working backwards, it is enough to prove that if T has no model then some finite subset of T has no model. If T has no model then T entails $\perp$, since $\perp$ has no models. So by the Compactness Theorem, some finite subset U of T entails $\perp$. But this implies that U has no model. QED

The next result is the weakest of a family of theorems known as the *Downward Löwenheim–Skolem Theorem.*

Theorem 1.8 Suppose L is a first-order language with at most countably many formulas, and let T be a consistent theory in L. Then T has a model with at most countably many elements.

Proof Assuming T is semantically consistent, it is $\mathcal{H}$-consistent by the correctness of $\mathcal{H}$. So the sketch proof of the Completeness Theorem in section 1.9 constructs a model A of T. By the last sentence of section 1.9, A has at most countably many elements. QED

There is also an *Upward Löwenheim–Skolem Theorem*, which says that every first-order theory with infinite models has arbitrarily large models.

A *basic conjunction* is a formula of the form

$(\phi_1 \,\&\, \cdots \,\&\, \phi_m)$

where each ϕ_i is either an atomic formula or an atomic formula preceded by $\sim$. (Note that $m = 1$ is allowed, so that a single atomic formula, with or without $\sim$, counts as a basic conjunction.) A formula is in *disjunctive normal form* if it has the form

$(\psi_1 \vee \cdots \vee \psi_n)$

where each ψ_j is a basic conjunction. (Again, $n = 1$ is allowed, so that a basic conjunction counts as being in disjunctive normal form.)

A first-order formula is said to be *prenex* if it consists of a string of quantifiers followed by a formula with no quantifiers in it. (The string of quantifiers may be empty, so that a formula with no quantifiers counts as being prenex.)

A formula is in *normal form* if it is prenex and the part after the quantifiers is in disjunctive normal form.

Theorem 1.9 (*Normal Form Theorem*) Every first-order formula ϕ is equivalent to a first-order formula ψ of the same signature as ϕ, which has the same free variables as ϕ and is in normal form.

The next theorem, Lyndon's Interpolation Theorem, deserves to be better known. Among other things, it is the first-order form of some laws which were widely known to logicians of earlier centuries as the Laws of Distribution (Hodges, 1998). It is stated here for sentences in normal form; by Theorem 1.9, this implies a theorem about all first-order sentences.

Suppose ϕ is a first-order sentence in normal form. An occurrence of a relation symbol in ϕ is called *positive* if it has no '$\sim$' immediately in front of it, and *negative* if it has.

Theorem 1.10 (*Lyndon's Interpolation Theorem*) Suppose ϕ and ψ are first-order sentences in normal form, and ϕ entails ψ. Then there is a first-order sentence θ in normal form, such that

- ϕ entails θ and θ entails ψ
- every relation symbol which has a positive occurrence in θ has positive occurrences in both ϕ and ψ, and
- every relation symbol which has a negative occurrence in θ has negative occurrences in both ϕ and ψ.

Lyndon's theorem can be proved either by analyzing proofs of the sequent '$\phi \vdash \psi$', or by a set-theoretic argument using models of ϕ and ψ. Both arguments are too complicated to give here.

An important corollary of Lyndon's Interpolation Theorem is Craig's Interpolation Theorem, which was proved a few years before Lyndon's.

Corollary 1.11 (*Craig's Interpolation Theorem*) Suppose ϕ and ψ are first-order sentences, and ϕ entails ψ. Then there is a first-order sentence θ such that

- ϕ entails θ and θ entails ψ
- every relation symbol that occurs in θ occurs both in ϕ and in ψ.

Craig's Interpolation Theorem in turn implies Beth's Definability Theorem, which was proved earlier still. But all these theorems are from the 1950s, perhaps the last great age of elementary metatheory.

Corollary 1.12 (*Beth's Definability Theorem*) Suppose ϕ is a first-order sentence in which a relation symbol R of arity n occurs, and suppose also that there are not two models I and J of ϕ which are identical except that R^I is different from R^J. Then ϕ entails some first-order sentence of the form

$$(\forall x_1) \cdots (\forall x_n) \, (\psi \equiv R(x_1, \ldots, x_n))$$

where ψ is a formula in which R never occurs.

Finally, note a metatheorem of a different kind, to contrast with Theorem 1.2 above: a form of *Church's Theorem on the Undecidability of First-Order Logic*.

Theorem 1.13 Suppose L is a first-order language whose signature contains at least one n-ary relation symbol with $n > 1$. Then L is not decidable.

A reference for all the metatheorems in this section except Theorems 1.9 and 1.13 is Hodges (1997). Theorem 1.9 is proved in both Kleene (1952, pp. 134f, 167) and Ebbinghaus et al. (1984, p. 126), together with a wealth of other mathematical information about first-order languages. Boolos and Jeffrey (1974) contains a proof of the undecidability of first-order logic (though to reach Theorem 1.13 above from its results, some coding devices are needed).

Suggested further reading

There are many places where the subjects of this chapter can be pursued to a deeper level. Of those mentioned already in this chapter, Boolos and Jeffrey (1974) is a clear introductory text aimed at philosophers, while Hodges (1983) is a survey with an eye on philosophical issues. Ebbinghaus et al. (1984) is highly recommended for those prepared to face some nontrivial mathematics. Of older books, Church (1956) is still valuable for its philosophical and historical remarks, and Tarski (1983) is outstanding for its clear treatment of fundamental questions.

References

Barwise, J. and Etchemendy, J. 1999: *Tarski's World 5.1*, available with *Language, Proof and Logic*, (Seven Bridges Press, New York and London).

Boolos, G. S. and Jeffrey, R. C. 1974: *Computability and Logic*, (Cambridge University Press, Cambridge).
Church, A. 1956: *Introduction to Mathematical Logic*, (Princeton University Press, Princeton, NJ).
Ebbinghaus, H.-D., Flum, J. and Thomas, W. 1984: *Mathematical Logic*, (Springer, New York).
Gallier, J. H. 1986: *Logic for Computer Science: Foundations of Automated Theorem Proving*, (Harper and Row, New York).
Gentzen, G. 1969: "Untersuchungen über das Logische Schliessen," translated in *The Collected Papers of Gerhard Gentzen*, M. Szabo, ed., (North-Holland, Amsterdam), 68–131.
Hilbert, D. and Ackermann, W. 1950: *Grundzuge der Theoretische Logik* (1928); English translation *Principles of Mathematical Logic*, R. E. Luce, ed., (Chelsea Publishing Company, New York).
Hintikka, J. 1973: *Logic, Language-Games and Information*, (Oxford University Press, Oxford).
Hodges, W. 1983: "Elementary Predicate Logic," in *Handbook of Philosophical Logic – Vol. I: Elements of Classical Logic*, D. M. Gabbay and F. Guenthner, eds., (D. Reidel, Dordrecht), 1–131.
Hodges, W. 1997: *A Shorter Model Theory*, (Cambridge University Press, Cambridge).
Hodges, W. 1998: "The Laws of Distribution for Syllogism," *Notre Dame Journal of Formal Logic*, 39, 221–30.
Kleene, S. C. 1952: *Introduction to Metamathematics*, (North-Holland, Amsterdam).
Sundholm, G. 1983: "Systems of Deduction," in *Handbook of Philosophical Logic – Vol. I: Elements of Classical Logic*, D. M. Gabbay and F. Guenthner, eds., (D. Reidel, Dordrecht), 133–88.
Tarski, A. 1983: "The Concept of Truth in Formalized Languages," in *Logic, Semantics, Metamathematics, Papers from 1923 to 1938*, J. H. Woodger, tr., and J. Corcoran, ed., (Hackett Publishing Co., Indianapolis), 152–278.
Tarski, A. and Vaught, R. 1957: "Arithmetical Extensions of Relational Systems," *Compositio Math.*, 13, 81–102.