Chapter 2
Propositional- and Predicate-Calculus Preliminaries

This chapter prepares for the extensive account of our verifier system given in Chap. 4 by describing and analyzing two of the system's basic ingredients, the *propositional calculus*, from which we take all necessary properties of the logical operations &, $\vee$, $\neg$, $\rightarrow$, and $\leftrightarrow$, and the (first-order) *predicate calculus*, which to these propositional mechanisms adds compound functional and predicate constructions and the two quantifiers $\forall$ and $\exists$. Then we will show the axioms of a classical specification of *set theory* in predicate calculus; to end, we will highlight the much-debated issue of the consistency of this Zermelo–Fraenkel–Skolem theory and of some of its proposed extensions.

Why Predicate Calculus? Our aim is to develop a mechanism capable of ensuring that the logical formulae in which we are interested are universally valid. Since, as we shall see in Chap. 6, there can exist no algorithm capable of making this determination in all cases, we must use the mechanism of *proof*. This embeds the formulae in which we are interested in some system of sequences of formulae, within which we can define a property ls_a_proof(p) capable of being verified by an algorithm, such that we can be certain that the final component t of any sequence p satisfying ls_a_proof(p) is universally valid. Then we can use intuition freely to find aesthetically pleasing sequences p, the proofs, leading to interesting end goals t, the *theorems*. In principle, any system of formulae and sequences of formulae having this property is acceptable. The propositional/predicate calculus and set theory in which we work is merely one such formalism, of interest because of its convenience and wide use, and because much effort has gone into ensuring its reliability.

2.1 The Propositional Calculus

The propositional calculus constitutes the 'bottom-most' part of the full logical formalism with which we will work in this book. It provides only the operations &, $\vee$, $\neg$, $\rightarrow$, and $\leftrightarrow$ and the two constants 'true' and 'false', all other symbolic con-

J.T. Schwartz et al., *Computational Logic and Set Theory*,
DOI 10.1007/978-0-85729-808-9_2,

structions being reduced ('blobbed') down to single letters when propositional deductions must be made. An example given earlier, i.e. the formula

$$\big(F(x+y) = F\big(F(x)\big) \to F\big(F(x)\big) = 0\big) \to \big(F\big(F(x)\big) \neq 0 \to F(x+y) \neq F\big(F(x)\big)\big)$$

whose 'blobbed' propositional skeleton is

$$(p \to q) \to \big((\neg q) \to (\neg p)\big),$$

illustrates what is meant.

Formulae of the propositional calculus are built starting with string names designating propositional variables and combining them using the dyadic infix operators '&', '$\vee$', '$\to$', and '$\leftrightarrow$' and the monadic operator '$\neg$'. Parentheses are used to group the subparts of formulae. The only precedence relation supported is the rule that '&' binds more tightly than '$\vee$', so parentheses must normally be used rather liberally. Syntactically, the propositional calculus is a simple operator language, whose (syntactically valid) formulae parse unambiguously into syntax trees, each of whose internal nodes is marked either with one of the allowed infix operators, in which case it has two descendants, or with the monadic operator '$\neg$', in which case it has one descendant. Each leaf of such a tree is marked either with the name of a propositional variable or with one of the two allowed constant symbols 'true' and 'false'.

An example is

$$(\mathsf{pan} \to \mathsf{quack}) \to \big((\neg\mathsf{quack}) \to (\neg\mathsf{true})\big).$$

Here the propositional variables which appear are 'pan' and 'quack', and the constant 'true' also appears.

Since the derivation of the syntax tree of a propositional formula from its string form ('parsing') and of the string form from the syntax tree ('unparsing') are both standard programming operations, we generally regard these two structures as being roughly synonymous and use whichever is convenient without further ado.

As in other logical systems we can think of our formulae either in terms of the values of functions which they represent, or as statements deducible from one another under certain circumstances, and so as the ingredients of some system of formalized proof. We begin with the first approach. In this way of looking at things, each propositional variable represents one of the truth values 1 or 0, which the propositional operators combine in standard ways. The following more formal definition captures this idea:

Definition 2.1 An *assignment* for a collection of propositional formulae is a single-valued function A mapping each of its constants and variables into one of the two values 1 and 0. Each assignment is required to map 'true' into 1 and 'false' into 0. The assignment is said to *cover* each of the formulae in the collection.

Given any such assignment A, and a formula F which it covers, the value $\mathsf{Val}(A, F)$ *of the assignment A for the expression F* is the Boolean value defined in the following recursive way.

(i) If the formula F is just a variable x or is one of the constants 'true' and 'false', then $\mathsf{Val}(A, F) = A(F)$.
(ii) If the formula F has the form '$G \,\&\, H$', then $\mathsf{Val}(A, F)$ is the minimum of $\mathsf{Val}(A, G)$ and $\mathsf{Val}(A, H)$.
(iii) If the formula F has the form '$G \vee H$', then $\mathsf{Val}(A, F)$ is the maximum of $\mathsf{Val}(A, G)$ and $\mathsf{Val}(A, H)$.
(iv) If the formula F has the form '$\neg G$', then $\mathsf{Val}(A, F) = 1 - \mathsf{Val}(A, G)$.
(v) If the formula F has the form '$G \to H$', then $\mathsf{Val}(A, F) = \mathsf{Val}(A,\ \text{`}(\neg G) \vee H\text{'})$.
(vi) If the formula F has the form '$G \leftrightarrow H$', then

$$\mathsf{Val}(A, F) = \mathsf{Val}\big(A,\ \text{`}(G \,\&\, H) \vee \big((\neg G) \,\&\, (\neg H)\big)\text{'}\big).$$

Definition 2.2 A propositional formula F is a *tautology* if $\mathsf{Val}(A, F) = 1$ for all the assignments A covering it.

So tautologies are propositional formulae which evaluate to true no matter what truth values are assigned to their variables. Examples are

$$p \vee (\neg p), \qquad q \to (p \to q), \qquad p \to \big(q \to (p \,\&\, q)\big),$$

and many others, some listed below. These are the propositional formulae which possess 'universal logical validity'.

Since the number of possible assignments A for a propositional formula F is at most 2^n, where n is the number of variables in the formula, we can determine whether F is a tautology by evaluating $\mathsf{Val}(A, F)$ for all such A. An alternative approach is to establish a system of proof by singling out some initial collection of tautologies (which we will call 'axioms') from which all remaining tautologies can be derived using rules of inference, which must also be defined. (This is the 'logical system' approach.) The axioms and rules of inference can be chosen in many ways. Though not at all the smallest possible set, the following collection has a familiar and convenient algebraic flavor.

(i) $(p \,\&\, q) \leftrightarrow (q \,\&\, p)$
(ii) $((p \,\&\, q) \,\&\, r) \leftrightarrow (p \,\&\, (q \,\&\, r))$
(iii) $(p \,\&\, p) \leftrightarrow p$
(iv) $(p \vee q) \leftrightarrow (q \vee p)$
(v) $((p \vee q) \vee r) \leftrightarrow (p \vee (q \vee r))$
(vi) $(p \vee p) \leftrightarrow p$
(vii) $(\neg(p \,\&\, q)) \leftrightarrow ((\neg p) \vee (\neg q))$
(viii) $(\neg(p \vee q)) \leftrightarrow ((\neg p) \,\&\, (\neg q))$
(ix) $((p \vee q) \,\&\, r) \leftrightarrow ((p \,\&\, r) \vee (q \,\&\, r))$
(x) $((p \,\&\, q) \vee r) \leftrightarrow ((p \vee r) \,\&\, (q \vee r))$
(xi) $(p \leftrightarrow q) \to ((p \,\&\, r) \leftrightarrow (q \,\&\, r))$
(xii) $(p \leftrightarrow q) \to ((p \vee r) \leftrightarrow (q \vee r))$

(xiii) $(p \leftrightarrow q) \rightarrow ((\neg p) \leftrightarrow (\neg q))$
(xiv) $(p \leftrightarrow q) \rightarrow (q \rightarrow p)$
(xv) $(p \rightarrow q) \leftrightarrow ((\neg p) \vee q)$
(xvi) $(p \leftrightarrow q) \leftrightarrow ((p \rightarrow q)\ \&\ (q \rightarrow p))$
(xvii) $(p\ \&\ q) \rightarrow p$
(xviii) $(p \leftrightarrow q) \rightarrow ((q \leftrightarrow r) \rightarrow (p \leftrightarrow r))$
(xix) $(p \leftrightarrow q) \rightarrow (q \leftrightarrow p)$
(xx) $(p \leftrightarrow p)$
(xxi) $(p\ \&\ (\neg p)) \leftrightarrow \mathsf{false}$
(xxii) $(p \vee (\neg p)) \leftrightarrow \mathsf{true}$
(xxiii) $(\neg(\neg p)) \leftrightarrow p$
(xxiv) $(p\ \&\ \mathsf{true}) \leftrightarrow p$
(xxv) $(p\ \&\ \mathsf{false}) \leftrightarrow \mathsf{false}$
(xxvi) $(p \vee \mathsf{true}) \leftrightarrow \mathsf{true}$
(xxvii) $(p \vee \mathsf{false}) \leftrightarrow p$
(xxviii) $(\neg\mathsf{true}) \leftrightarrow \mathsf{false}$
(xxix) $(\neg\mathsf{false}) \leftrightarrow \mathsf{true}$
(xxx) true

The preceding are to be understood as axiom 'templates' or 'schemas', in the sense that all formulae resulting from one of them by substitution of syntactically legal propositional formulae $P, Q, \ldots$ for the letters $p, q, \ldots$ occurring in them are also axioms. For example,

$$\big(\big((p \vee q) \vee (r \rightarrow r)\big)\ \&\ \big((p \vee q) \vee (r \rightarrow r)\big)\big) \leftrightarrow \big((p \vee q) \vee (r \rightarrow r)\big)$$

is a substituted instance of (iii) and therefore is also regarded as an axiom.

The reader can verify that all of the axioms listed are in fact tautologies.

In the presence of this lush collection of axioms we need only one rule of inference (namely the '*modus ponens*' of mediaeval logicians). From any two formulae of the form p and $p \rightarrow q$ this allows us to deduce q. As with the axioms, this rule is to be understood as a template, covering all of its substituted instances.

To ensure that the tautologies are exactly the derivable propositional formulae we must prove *soundness*, namely that (I) only tautologies can be derived, and *completeness*, namely that (II) all tautologies can be derived. (I) is easy. We reason as follows. All the axioms are tautologies. Moreover, since

$$\mathsf{Val}(A,\ p \rightarrow q) = \max\big(1 - \mathsf{Val}(A, p),\ \mathsf{Val}(A, q)\big),$$

it follows that if $\mathsf{Val}(A, p \rightarrow q)$ and $\mathsf{Val}(A, p)$ are both 1, so is $\mathsf{Val}(A, q)$. So if '$p \rightarrow q$' and p are both tautologies, then so is q. This proves our claim (I).

Proving claim (II) takes a bit more work, whose general pattern is much like that used to reduce multivariate polynomials to their canonical form. Starting with any syntactically well-formed propositional formula F, we can proceed in the following way to derive a chain of formulae equivalent to F (via an explicit chain of equivalences $F_i \leftrightarrow F_{i+1}$). Note that axioms (xviii–xx) ensure that the equivalence relator

'$\leftrightarrow$' has the same transitivity, symmetry, and reflexivity properties as equality, while (xi–xiii) allow us to replace any subexpression of an expression formed using only the three operators &, $\vee$, $\neg$ by any equivalent subexpression.

Using these facts and (xv–xvi) we first descend recursively through the syntax tree of F, replacing any occurrence of one of the operations $\rightarrow$, $\leftrightarrow$ by an equivalent expression involving only &, $\vee$, $\neg$. This reduces F to an equivalent formula involving only the operators &, $\vee$, $\neg$. Then, using (vii–viii) and (x), we systematically push '$\neg$' and '$\vee$' operators down in the syntax tree, moving '&' operators up. Subformulae of the form $(\neg(\neg p))$ are simplified to p using axiom (xxiii). Axioms (xxiv–xxix) can be used to simplify expressions containing the constants 'true' and 'false'. When this work is complete F will been have reduced to an equivalent formula F' which is either one of the constants 'true' or 'false' or has the form $a_1 \,\&\, \cdots \,\&\, a_k$, where each a_j is a disjunction of the form

$$b_1 \vee \cdots \vee b_h,$$

each b_m being either a propositional variable or the negation of a propositional variable. (ii) and (v) allow us to think of these conjunctions and disjunctions without worrying about how they are parenthesized. Then (iv) and (vi) can be used to bring all the b_m involving a particular propositional variable together within each a_j.

Now assume that F is a tautology, so that every one of the formulae to which we have reduced it must also be a tautology (since the substitutions performed all convert tautologies to tautologies), and so our final formula F' is a tautology. We will now further reduce F', so that it becomes the formula 'true'. Unless F' is already 'true', in each a_j, there must occur at least one pair b_m, b_n of disjuncts such that b_m is a propositional variable of which b_n is the negation, '$\neg b_m$'. Indeed, if this is not the case, then any propositional variable which occurs in a_j will occur either negated or non-negated, but not both. Given this, we can assign the value 0 to each non-negated variable and the value 1 to each negated variable. Then every b_m in a_j will evaluate to 0, so the whole expression $b_1 \vee \cdots \vee b_h$ will evaluate to 0, that is, a_j will evaluate to 0. But as soon as this happens the whole formula $a_1 \,\&\, \cdots \,\&\, a_k$ will evaluate to 0. This shows that there exists an assignment A such that $\mathsf{Val}(A, F') = 0$, contradicting the fact that F' is a tautology. This contradiction proves our claim that each a_j must contain at least one pair b_m, b_n of disjuncts which agree except for the presence of a negation operator in one but not in the other.

Given this fact, (xxii) tells us that '$b_m \vee b_n$' simplifies to 'true', so that (xxvi) can be used repeatedly to simplify a_j to 'true'. Since this is the case for each a_j, repeated use of (xxiv) allows us to reduce any tautology to 'true' using a chain of equivalences. Since this chain of equivalences can as well be traversed in the reverse direction, we can equally well expand the axiom 'true' (axiom (xxx)) into our original formula F using a chain of equivalences. Then (xiv) can be used to convert this chain of equivalences into a chain of implications, giving us a proof of F, by repeated uses of modus ponens.

Any set of axioms from which all the statements (i–xxx) can be derived as theorems can clearly be used as an axiomatic basis for the propositional calculus. This

allows much leaner sets of axioms to be used. We refrain from exploring this point, which lacks importance for the rest of our discussion.

However, it is worth embedding the notion of 'tautology' in a wider, relativized, set of ideas. Suppose that we write

$$\models F$$

to indicate that the formula F is a tautology, and

$$\vdash F$$

to indicate that F is a provable formula of the propositional calculus. The preceding discussion shows that $\models F$, and $\vdash F$, are equivalent conditions. This result can be generalized as follows. Let S designate any finite set of syntactically well-formed formulae of the propositional calculus. We can then write

$$S \models F$$

to indicate that, for each assignment A covering both F, and all the formulae in S, we have $\mathsf{Val}(A, F) = 1$ whenever $\mathsf{Val}(A, G) = 1$ for all G in S. Also, we write

$$S \vdash F$$

to indicate that F follows by propositional proof if the statements in S are added to the axioms of propositional calculus (each of them acting as an individual axiom, not as a template). Then it is easy to show that

$$S \models F \quad \text{if and only if} \quad S \vdash F.$$

To show this, first suppose that $S \models F$. Let C designate the conjunction

$$G_1 \,\&\, \cdots \,\&\, G_k$$

of all the formulae in S. Then since $\mathsf{Val}(A, H_1 \,\&\, H_2) = \min(\mathsf{Val}(A, H_1), \mathsf{Val}(A, H_2))$ for any two formulae H_1, H_2, it follows that $\mathsf{Val}(A, C) = 1$ if and only if $\mathsf{Val}(A, G) = 1$ for all G in S. We have

$$\mathsf{Val}(A,\ C \to F) = \mathsf{Val}\big(A,\ (\neg C) \vee F\big) = \max\big(1 - \mathsf{Val}(A, C),\ \mathsf{Val}(F)\big)$$

for all assignments A covering $C \to F$, (i.e. covering both F, and all the formulae in S). It follows that for each assignment A covering both F, and all the formulae in S, we have $\mathsf{Val}(A, C \to F) = 1$, since if $1 - \mathsf{Val}(A, C) \neq 1$ then $\mathsf{Val}(A, C)$ must be 1 and so $\mathsf{Val}(F)$ must be 1. Thus

$$\models C \to F,$$

and so it follows that $\vdash C \to F$, i.e. $C \to F$ can be proved from the axioms of propositional calculus alone. But then if the statements in S are added as additional

axioms we can prove F, by first proving $C \to F$, and then using the statements in S to prove the conjunction C. This shows that $S \models F$ implies $S \vdash F$.

Next suppose that $S \vdash F$, and let A be an assignment covering both F, and all the formulae in S so that $\mathsf{Val}(A, G) = 1$ for every statement G in S. Then $\mathsf{Val}(A, G) = 1$ for every statement G that can be used as an axiom in the proof of F, from the standard axioms of propositional calculus and the statements in S as additional axioms. But we have seen above that if $\mathsf{Val}(A, p \to q)$ and $\mathsf{Val}(A, p)$ are both 1, so is $\mathsf{Val}(A, q)$. Since derivation of q from p and $p \to q$ is the only inference step allowed in propositional calculus proofs, it follows that $S \models F$, completing our proof that the conditions $S \models F$, and $S \vdash F$, are equivalent.

We shall see that similar statements apply to the much more general predicate calculus studied in the following section. In that section, we will need the following extension of the preceding results to countably infinite collections of propositional formulae.

Definition 2.3 A (finite or infinite) collection S of formulae of the propositional calculus is said to be *consistent* if the proposition 'false' cannot be deduced from S, i.e.

$$S \vdash \mathsf{false}$$

is false. We say that S *has a model* A if there exists some assignment A covering all the formulae of S such that $\mathsf{Val}(A, F) = 1$ for every F in S.

Theorem 2.1 (Compactness) *Let S be a denumerable collection of formulae of the propositional calculus. Then the following three conditions are equivalent*:

(i) *S is consistent.*
(ii) *Every finite subset of S is consistent.*
(iii) *S has a model.*

Proof Since subsets of a consistent S are plainly consistent, (i) implies (ii). On the other hand, any proof of 'false' from the statements of S is of finite length by definition, and so uses only a finite number of the statements of S. Thus (ii) implies (i), so (ii) and (i) are equivalent.

Next suppose that S is not consistent, so that 'false' can be proved from some finite subset S' of the statements in S. Let C be the conjunction of all the statements in S'. It follows from the discussion immediately preceding the statement of the present theorem that $\vdash C \to \mathsf{false}$, and so $\mathsf{Val}(A, `C \to \mathsf{false}') = 1$ for any assignment A covering all the propositional symbols in S. This gives $\mathsf{Val}(A, C) = 0$ for all such A, so that S has no model. This proves that (iii) implies (i).

Next we show that (i) implies (iii). For this, let $\{S_j\}$ be an increasing sequence of finite subsets of S whose union is all of S. Each S_j is plainly consistent, so

$$S_j \vdash \mathsf{false}$$

is false for each j, and therefore

$$S_j \models \mathsf{false}$$

is false, since we have shown above that these two conditions are equivalent for finite S_j. That is, for each j there must exist an assignment A_j covering all the variables appearing in any formula of S_j, such that $\mathsf{Val}(A_j, S_j) = 1$. Let $v_1, v_2, v_3, \ldots$ be an enumeration of all the variables appearing in any of the formulae of S. Then each v_k must be in the domain of all A_j for all j beyond a certain point $j = j_k$.

Let I_0 designate the sequence of all integers. Since $A_j(v_1)$ must have one of the two values 0 and 1, there must exist an infinite subsequence I_1 of I_0 for all j of which $A_j(v_1)$ has the same value. Call this value $B(v_1)$. Arguing in the same way we see that here must exist an infinite subsequence I_2 of I_1 and a Boolean value $B(v_2)$ such that

$$B(v_2) = A_j(v_2) \quad \text{for all } j \text{ in } I_2.$$

Arguing repeatedly in this way we eventually construct values $B(v_k)$ for each k such that for each finite m, there exist infinitely many j such that

$$B(v_n) = A_j(v_n) \quad \text{for all } n \text{ from 1 to } m.$$

Now consider any of the formulae G of S. Since G can involve only finitely many propositional variables v_j, all its variables will be included in the set $\{v_1, \ldots, v_k\}$ for each sufficiently large k. Take any A_j for which $B(v_n) = A_j(v_n)$ for all n from 1 to k. Then it is clear that for some i greater than j, we have

$$\mathsf{Val}(B, G) = \mathsf{Val}(A_i, G) = 1.$$

Hence $\mathsf{Val}(B, G) = 1$ for all G in S, so that B is a model of S, proving that (i) implies (iii), and thereby completing the proof of our theorem. □

Using the Compactness Theorem, we can show that the conditions $S \vdash F$, and $S \models F$, are equivalent even in the case in which S is an infinite set of propositional formulae.

To show this, first assume that $S \models F$. Then the set $S \cup \{\neg F\}$ of propositions is plainly not consistent, and so by the Compactness Theorem S must contain some finite subset S_0 such that $S_0 \cup \{\neg F\}$ is not consistent. Then plainly $S_0 \models F$, so we have $S_0 \vdash F$. This clearly implies $S \vdash F$; so $S \vdash F$, follows from $S \models F$.

But, as noted at the end of the proof of the Compactness Theorem, $S \models F$ follows from $S \vdash F$, even if S is infinite, completing the proof of our claim.

2.2 The Predicate Calculus

The predicate calculus constitutes the next main part of the logical formalism used in this book. This calculus enlarges the propositional calculus, preserving all its

operations but also allowing compound functional and predicate terms and the two quantifiers $\forall$ and $\exists$. An example is the formula

$$\big((\forall x, y \mid F(x+y) = F\big(F(x)\big)\big) \to F\big(F(x)\big) = 0\big) \\ \to \big((\exists x \mid F\big(F(x)\big) \neq 0\big) \to \big(F(x+y) \neq F\big(F(x)\big)\big)\big).$$

Formulae of the predicate calculus are built starting with string names of three kinds, respectively, designating 'individual' variables, function symbols, and predicate symbols. These are combined into '*terms*', '*atomic formulae*', and '*formulae*' using the following recursive syntactic rules.

(i) Any variable name is a term. (We assume variable names to be alphanumeric and to start with lower case letters.)

(ii) Each function symbol has some fixed finite number k of arguments. If f is a function symbol of k arguments, and $t_1, \ldots, t_k$ are any k terms, then $f(t_1, \ldots, t_k)$ is a term. (We assume function names to be alphanumeric and to start with lower case letters.)

(iii) Each predicate symbol has some fixed finite number k of arguments. If P is a predicate symbol of k arguments, and $t_1, \ldots, t_k$ are any k terms, then $P(t_1, \ldots, t_k)$ is an atomic formula. (We assume predicate names to be alphanumeric and to start with upper case letters.)

(iv) Formulae are formed starting from atomic formulae and using the operators and syntactic rules of the propositional calculus and the two quantifiers $\forall$ and $\exists$. More precisely, if e and f are any two predicate formulae and $v_1, \ldots, v_n$ are any n variable names, with $n > 0$, then the following expressions are predicate formulae:

$$\begin{array}{lll} e \,\&\, f, & e \vee f, & \neg e, \\ e \to f, & e \leftrightarrow f, & \\ (\forall v_1, \ldots, v_n \mid e), & (\exists v_1, \ldots, v_n \mid e). & \end{array}$$

Like propositional formulae, the formulae of predicate calculus parse unambiguously into syntax trees each of whose internal nodes is marked either (i) with one of the propositional operators, and then has as many descendants as the corresponding propositional node, or (ii) with a function or predicate symbol, in which case its descendants correspond to the arguments of the function or predicate symbol; (iii) a quantifier $\forall$ or $\exists$ involving n variable names, in which case the node has $n+1$ descendants, the first n marked with the n variable names appearing in the quantifier and the $n+1$-st which is the syntax tree of the expression e that is being quantified. Each leaf of such a tree is marked either with the name of an individual variable or a function symbol of zero arguments. (Such function symbols are called 'constants'.)

Each occurrence of a variable v at a leaf of the syntax tree of a valid predicate formula is either *free* or *bound*. A variable v is considered to be bound if it appears as the descendant of some syntax tree node which is marked with a quantifier in

whose associated list of variables v occurs; otherwise the occurrence is a free occurrence. These notions clearly translate back into corresponding notions for variable occurrences in the unparsed string forms of the same formulae. For example, in the predicate formula

$$\big(\forall x, z, x \mid F(x + y + z)\big) \vee \big(\exists y, y \mid F(x + y)\big)$$

the first three occurrences of x are bound, but the fourth occurrence of x is free. Likewise the last three occurrences of y are bound, but its first occurrence is free. Note that, as this example shows, repeated occurrences of a variable in the list following one of the quantifier symbols $\forall$ or $\exists$ are legal. However, we will see, when we come to define the semantics of predicate formulae, that such repetitions are always superfluous since any variable occurrence repeated later in the list following a quantifier symbol can simply be dropped. For example, the formula shown above has the same meaning as

$$\big(\forall z, x \mid F(x + y + z)\big) \vee \big(\exists y \mid F(x + y)\big).$$

Bound variables are considered to belong to the *scope* of the nearest ancestor quantifier in whose list of variables they appear; this quantifier is said to *bind* them. For example, in

$$\big(\forall x \mid F(x) \vee \big(\exists x \mid G(x)\big) \vee H(x)\big)$$

the first, second, and final occurrences of x are in the scope of the first quantifier '$\forall$', but the third and fourth occurrences are in the scope of the second quantifier '$\exists$'.

As was the case for the propositional calculus, we can think of predicate formulae either as representing certain functions, or as the ingredients of a system of formalized proof. Again we begin with the first approach. Here the required definitions are a bit trickier.

Definition 2.4 An *interpretation framework* for a collection PF of predicate formulae is a triple $(\mathscr{U}, I, A)$ such that

(i) $\mathscr{U}$ is a nonempty set, called the *universe* or *domain* of the interpretation framework. We write $\mathscr{U}^k$ for the k-fold Cartesian product of $\mathscr{U}$ with itself.

(ii) I is a single-valued function, called an *interpretation*, which maps each of the function and predicate symbols occurring in the collection in accordance with the following rules:
  (ii.a) Each function symbol f of k arguments occurring in the collection of formulae is mapped into a function $I(f)$ which sends $\mathscr{U}^k$ into $\mathscr{U}$.
  (ii.b) Each predicate symbol P of k arguments occurring in the collection of formulae is mapped into a function $I(P)$ which sends $\mathscr{U}^k$ into the set $\{0, 1\}$ of values.

(iii) A is a single-valued function, called an *assignment*, which maps each of the individual variables occurring freely in the collection PF of formulae into an element of $\mathscr{U}$.

As previously we speak of such an interpretation framework as *covering* the collection PF of predicate formulae.

Suppose that we are given any such interpretation I and assignment A with universe $\mathscr{U}$, and an expression F which they cover. (Note that F can be either a term or a predicate formula.) Then the value $\mathsf{Val}(I, A, F)$ *of the assignment for the expression* is the value defined in the following recursive way.

(i) If F is just an individual variable x, then $\mathsf{Val}(I, A, F) = A(x)$.

(ii) If F is a term having the form $g(t_1, \ldots, t_k)$, and G is the corresponding mapping $I(g)$ from $\mathscr{U}^k$ to $\mathscr{U}$, then $\mathsf{Val}(I, A, F) = G(\mathsf{Val}(I, A, t_1), \ldots, \mathsf{Val}(I, A, t_k))$.

(iii) If F is an atomic formula having the form $P(t_1, \ldots, t_k)$, and p is the corresponding mapping $I(P)$ from $\mathscr{U}^k$ to $\{0, 1\}$, then $\mathsf{Val}(I, A, F)$ is the 0/1 value $p(\mathsf{Val}(I, A, t_1), \ldots, \mathsf{Val}(I, A, t_k))$.

(iv) If F is a formula having the form $G \,\&\, H$, then $\mathsf{Val}(I, A, F)$ is the minimum of $\mathsf{Val}(I, A, G)$ and $\mathsf{Val}(I, A, H)$.

(v) If F is a formula having the form $G \vee H$, then $\mathsf{Val}(I, A, F)$ is the maximum of $\mathsf{Val}(I, A, G)$ and $\mathsf{Val}(I, A, H)$.

(vi) If F is a formula having the form $\neg G$, then $\mathsf{Val}(I, A, F) = 1 - \mathsf{Val}(I, A, G)$.

(vii) If F is a formula having the form $G \to H$, then $\mathsf{Val}(I, A, F) = \mathsf{Val}(I, A, \text{`}(\neg G) \vee H\text{'})$.

(viii) If F is a formula having the form $G \leftrightarrow H$, then $\mathsf{Val}(I, A, F) = \mathsf{Val}(I, A, \text{`}(G \,\&\, H) \vee ((\neg G) \,\&\, (\neg H))\text{'})$.

(ix) If F is a formula having the form $(\forall v_1, \ldots, v_n \mid e)$, then $\mathsf{Val}(I, A, F)$ is the minimum of $\mathsf{Val}(I, A', e)$, extended over all assignments A' such that A' covers the formula e and $A'(x) = A(x)$ for every variable x not in the list $v_1, \ldots, v_n$.

(x) If F is a formula having the form $(\exists v_1, \ldots, v_n \mid e)$, then $\mathsf{Val}(I, A, F)$ is the maximum of $\mathsf{Val}(I, A', e)$, extended over all assignments A' such that A' covers the formula e and $A'(x) = A(x)$ for every variable x not in the list $v_1, \ldots, v_n$.

Since, as seen in (ix) and (x) above, the variables appearing in the lists following quantifier symbols '$\forall$' and '$\exists$' merely serve to mark occurrences of the same variables in the quantifier's scope as being 'bound' and hence subject to minimization/maximization when values $\mathsf{Val}(I, A, F)$ are calculated, it follows that these variables can be replaced with any others provided that this replacement is made uniformly over the entire scope of each quantifier, and that no variable occurring freely in the original formula thereby becomes bound. For example, the formula

$$\big(\forall x \mid F(x) \vee \big(\exists x \mid G(x)\big) \vee H(x)\big)$$

appearing above can as well be written as

$$\big(\forall x \mid F(x) \vee \big(\exists y \mid G(y)\big) \vee H(x)\big)$$

or as

$$\big(\forall y \mid F(y) \vee \big(\exists x \mid G(x)\big) \vee H(y)\big).$$

A convenient way of performing this kind of 'bound variable standardization' is as follows. We make use of some standard list L of bound variable names, reserved for this purpose and used for no other. We work from the leaves of a formula's syntax tree up toward its root, processing all quantifiers more distant from the root before any quantifier closer to the root is processed. Suppose that a quantifier like

$$(\forall v_1, \ldots, v_n \mid e)$$

or

$$(\exists v_1, \ldots, v_n \mid e)$$

is encountered at a tree node Q during this process. We then take the first n variables $b_1, \ldots, b_n$ from the list L that do not already appear in any descendant of the node Q, replace $v_1, \ldots, v_n$ by $b_1, \ldots, b_n$, respectively, and make the same replacements for every free occurrence of any of the $v_1, \ldots, v_n$ in e.

This standardization will for example transform

$$\big(\forall y \,\big|\, \big(\forall y \,\big|\, F(y) \vee \big(\exists x \mid G(x)\big)\big) \vee H(y)\big)$$

into

$$\big(\forall b_3 \,\big|\, \big(\forall b_1 \,\big|\, F(b_1) \vee \big(\exists b_2 \mid G(b_2)\big)\big) \vee H(b_3)\big).$$

Such standardization of bound variables makes it easier to see what quantifier each bound variable occurrence relates to. It also uncovers identities between quantified subexpressions that might otherwise be missed, and so is a valuable preliminary to examination of the propositional structure of predicate formulae.

It also follows from (ix) and (x) that the value assigned to any quantified formula

$$(\forall v_1, v_2, \ldots, v_n \mid e) \tag{2.1}$$

is exactly the same as that assigned to

$$\big(\forall v_1 \,\big|\, \big(\forall v_2 \,\big|\, \big(\forall \cdots \mid (\forall v_n \mid e) \cdots\big)\big)\big) \tag{2.2}$$

and, likewise, the value assigned to any quantified formula

$$(\exists v_1, v_2, \ldots, v_n \mid e) \tag{2.3}$$

is exactly the same as that assigned to

$$\big(\exists v_1 \,\big|\, \big(\exists v_2 \,\big|\, \big(\exists \cdots \mid (\exists v_n \mid e) \cdots\big)\big)\big). \tag{2.4}$$

Accordingly, we shall regard (2.1) and (2.3) as abbreviations for (2.2) and (2.4). This allows us to assume (wherever convenient) that each quantifier examined in the following discussion involves only a single variable.

Definition 2.5 A predicate formula F is *universally valid* if $\mathsf{Val}(I, A, F) = 1$ for every interpretation framework $(\mathscr{U}, I, A)$ covering it.

In predicate calculus, universally valid formulae are those which evaluate to true no matter what 'meanings' are assigned to the variables, function symbols, and predicate symbols that occur within them. Examples are

$$P(x, y) \vee \big(\neg P(x, y)\big),$$
$$\big(\forall ya \mid Q(x) \to \big(P(x, y) \to Q(x)\big)\big),$$
$$\big(\forall x \mid P(x, y) \to \big(\exists y \mid \big(Q(x) \to \big(P(x, y)\ \&\ Q(x)\big)\big)\big)\big).$$

However, the problem of determining whether a given predicate formula is universally valid is of a much higher order of difficulty than the problem of recognizing propositional tautologies, since the collection of interpretation frameworks that must be considered is infinite rather than finite. There is no longer any reason for believing that this determination can be made algorithmically, and indeed it cannot, as we shall see in Chap. 6. Thus we have little alternative to setting up the predicate calculus as a logical system in which universally valid formulae are found by proof. We now begin to do this, starting with a special subclass of universally valid formulae, the *predicate tautologies*, which are defined as follows.

Definition 2.6 A predicate formula F is a *tautology* if it reduces to a propositional tautology by descending through its syntax tree and reducing each node not marked with a propositional operator to a single propositional variable, identical subnodes always being reduced to the same propositional variable. (In what follows we will call this latter formula the *propositional blobbing* of F.)

As an example, note that the indicated reduction sends

$$P(x, y) \vee \big(\neg P(x, y)\big) \text{ into } A \vee (\neg A),$$
$$\big(\forall y \mid Q(x) \to \big(P(x, y) \to Q(x)\big)\big) \text{ into } B,$$
$$P(x, y) \to \big(\exists y \mid \big(Q(x) \to \big(P(x, y)\ \&\ Q(x)\big)\big)\big) \text{ into } A \to C.$$

Thus the first of these three formulae is a predicate tautology, but the two others are not.

The recursive computation of $\mathsf{Val}(I, A, F)$ assigns some 0/1 value to each subtree of the syntax tree of F, and plainly assigns the same value to identical subtrees of the syntax tree of F. This makes it clear that every predicate tautology is universally valid. But there are other basic forms of universally well-formed predicate formulae, of which the most crucial are listed in the following definition.

Definition 2.7 A formula is an *axiom of the predicate calculus* if it is either

(i) any predicate tautology;
(ii) any formula of the form

$$\big((\forall y \mid P \to Q)\ \&\ (\forall y \mid P)\big) \to (\forall y \mid Q);$$

(iii) any formula of the form

$$\big(\neg(\forall y \mid \neg P)\big) \leftrightarrow (\exists y \mid P);$$

(iv) any formula of the form $P \leftrightarrow (\forall y \mid P)$, where the variable y does not occur in P as a free variable;

(v) any formula of the form $(\forall y \mid P) \rightarrow P(y \hookrightarrow e)$, where $P(y \hookrightarrow e)$ is the formula obtained from P by substituting the syntactically well-formed term e for each free occurrence of the variable y in P, provided that no variable free in e is bound at the point of occurrence of any such y in P.

We can easily see that all of these predicate axioms are universally valid. Given a formula P of the predicate calculus, let P' designate its propositional blobbing. Predicate tautologies are universally valid since the final stages of computation of $\mathsf{Val}(I, A, P)$ always use the values assigned to certain basic subformulae of P in the same way that values assigned to corresponding propositional variables are used in the propositional computation of $\mathsf{Val}(I, A, P')$. To see that (iii) is universally valid, we have only to note that for 0/1 valued functions f of any number of arguments we always have

$$\max(f) = 1 - \min(1 - f).$$

(iv) is universally valid because if y does not occur in P as a free variable, we have

$$\mathsf{Val}\big(I, A, \text{`}(\forall y \mid P)\text{'}\big) = \mathsf{Val}(I, A, P)$$

for every interpretation I and assignment A covering P.

(v) is universally valid because any interpretation I and assignment A covering $P(y \hookrightarrow e)$ will assign some value a_0 to e, and then $\mathsf{Val}(I, A, P(y \hookrightarrow e)) = \mathsf{Val}(I, A', P)$, where A' is the assignment identical to A except that it assigns the value a_0 to y. Since $\mathsf{Val}(I, A', (\forall y \mid P))$ is by definition the minimum of $\mathsf{Val}(I, B, P)$ extended over all assignments B which are identical to A except on the variable y, it follows that $\mathsf{Val}(I, A, \text{`}(\forall y \mid P)\text{'}) = 1$ implies $\mathsf{Val}(I, A, P(y \hookrightarrow e)) = 1$, so that

$$\max\big(1 - \mathsf{Val}\big(I, A, \text{`}(\forall y \mid P)\text{'}\big), \mathsf{Val}\big(I, A, P(y \hookrightarrow e)\big)\big)$$

is identically 1, i.e. $(\forall y \mid P) \rightarrow P(y \hookrightarrow e)$ is universally valid.

To show that (ii) is universally valid, note that for any interpretation I and assignment A covering (ii)

$$\mathsf{Val}\big(I, A, \text{`}(\forall y \mid P \rightarrow Q)\text{'}\big)$$

and

$$\mathsf{Val}\big(I, A, \text{`}(\forall y \mid P)\text{'}\big)$$

are, respectively, the minimum of $\max(1 - \mathsf{Val}(I, A', P), \mathsf{Val}(I, A', Q))$ and of $\mathsf{Val}(I, A', P)$, extended over all assignments A' which are identical to A except

on the variable y. If both of these minima are 1, then $1 - \mathsf{Val}(I, A', P)$ must be 0 for all such A', so $\mathsf{Val}(I, A', Q)$ must be 1 for all such A', proving that $\mathsf{Val}(I, A, \text{`}(\forall y \mid Q)\text{'}) = 1$. This implies the universal validity of (ii), completing our proof that all predicate axioms are universally valid.

2.2.1 Proof Rules of the Predicate Calculus

The predicate calculus has just two proof rules. The first is identical with the modus ponens rule of propositional calculus. The second is the *Rule of Generalization*, which states that if P is any previously proved result, then

$$(\forall x \mid P)$$

can be deduced.

A stronger variant of the Rule of Generalization, which turns out to be very useful in practice, allows us to deduce the formula

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

from $P \to Q$, provided that the variable x does not occur free in P. This variant can be justified as follows. Let us assume that the formula $P \to Q$ has been derived and that x is a variable which does not have free occurrences in P. By generalization and as instance of the predicate axiom (ii) we can derive the formulae

$$(\forall x \mid P \to Q), \quad \big((\forall x \mid P \to Q) \,\&\, (\forall x \mid P)\big) \to (\forall x \mid Q).$$

By propositional reasoning these imply the formula

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

Since we are assuming that the variable x does not occur free in P, we can derive the formula

$$P \leftrightarrow (\forall x \mid P)$$

using predicate axiom (iv), and it follows by propositional reasoning that

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

which establishes the strong form of the rule of generalization that we have stated.

In what follows we will not always distinguish between the two variants of the rule of generalization and we will use whichever version is more convenient for the purposes at hand. The argument given above shows that any proof which uses the strong variant of the Rule of Generalization can be transformed mechanically into a proof which uses only the standard form of this Rule.

We can easily see that any formula deduced from universally valid formulae using the two proof rules just explained must also be universally valid. For the modus ponens rule this follows as in the propositional case. For the rule of generalization we reason as follows. If $\mathsf{Val}(I, A, P) = 1$ for every interpretation I and assignment A covering P, then since for every assignment B covering $(\forall x \mid P)$ the value $v = \mathsf{Val}(I, B, \text{`}(\forall x \mid P)\text{'})$ is the minimum of $\mathsf{Val}(I, A, P)$ extended over all assignments A which give the same value as B to all variables other than x, it follows that $v = 1$ also.

In analogy with the case of the propositional calculus we write

$$\models F$$

to indicate that the formula F is a universally valid formula of the predicate calculus, and write

$$\vdash F$$

to indicate that F is a provable formula of the predicate calculus.

The following very important theorem is the predicate analog of the statement that a propositional formula is a tautology if and only if it is provable.

2.2.2 The Gödel Completeness Theorem

For any predicate formula, the conditions

$$\models F \quad \text{and} \quad \vdash F$$

are equivalent.

Half of this theorem is just as easy to prove as in the propositional case. Specifically, suppose that $\vdash F$. Then since all the axioms of predicate calculus are universally valid and the predicate-calculus rules of inference preserve universal validity, F must be universally valid, i.e. $\models F$.

The other, more difficult half of this theorem will be proved later, after some preparation. Much as in the case of the propositional calculus, this result can be generalized as follows. Let S designate any set of syntactically well-formed formulae of the predicate calculus. Write

$$S \models F$$

to indicate that, for each interpretation I and assignment A covering both F and all the formulae in S, we have $\mathsf{Val}(I, A, F) = 1$ whenever $\mathsf{Val}(I, A, G) = 1$ for all G in S. Also, write

$$S \vdash F$$

to indicate that F follows by predicate proof if the statements in S are added to the axioms of predicate calculus. Suppose that none of the formulae in S contain any

free variables (formulae with this property are usually called *sentences*). Then for any predicate formula, the conditions

$$S \models F \quad \text{and} \quad S \vdash F$$

are equivalent. (An easy example, given below, shows that we cannot omit the condition 'none of the formulae in S contain any free variables'.) The derivation of this from the more restricted result given by the Gödel completeness theorem is almost the same as the corresponding propositional proof. For the moment we will consider only the case in which S is finite. Suppose first that $S \models F$ and let C designate the conjunction

$$G_1 \,\&\, \cdots \,\&\, G_k$$

of all the formulae in S. Let I and A be, respectively, an interpretation and an assignment which cover $C \to F$ (i.e. cover both F and all the formulae in S). Then as in the propositional case it follows that $\mathsf{Val}(I, A, C) = 1$ if and only if $\mathsf{Val}(I, A, G) = 1$ for all G in S. Hence

$$\begin{aligned}\mathsf{Val}(I, A, C \to F) &= \mathsf{Val}\big(I, A, (\neg C) \vee F\big) \\ &= \max\big(1 - \mathsf{Val}(I, A, C), \mathsf{Val}(I, A, F)\big) = 1,\end{aligned}$$

for all such I and A. Hence

$$\models C \to F$$

follows using the Gödel Completeness Theorem, as stated above, and so it follows that

$$\vdash C \to F,$$

i.e. $C \to F$ can be proved from the axioms of predicate calculus alone. But then if the statements in S are added as additional axioms we can prove F by first proving $C \to F$, then using the statements in S to prove the conjunction C, and finally proving F by modus ponens from $C \to F$ and C. This shows that $S \models F$ implies $S \vdash F$.

Next suppose that there exists a formula F such that $S \vdash F$, but that $S \models F$ is false. Let F be such a formula with the shortest possible proof from S, and let I and A be, respectively, any interpretation and assignment A covering both F and all the formulae in S such that $\mathsf{Val}(I, A, G) = 1$ for every statement G in S, but $\mathsf{Val}(I, A, F) = 0$. The final step of a shortest proof of F from S cannot be either the citation of an axiom or the citation of a statement of S, since in both these cases we would have $\mathsf{Val}(I, A, F) = 1$. Hence this final step is either a modus ponens inference from two formulae p, $p \to F$ appearing earlier in the proof, or a generalization inference from one such formula p. In the modus ponens case we must have $S \models p$, $S \models p \to F$ by inductive assumption. Hence $\mathsf{Val}(I, A, p \to F)$ and $\mathsf{Val}(I, A, p)$ are both 1, and therefore so is $\mathsf{Val}(I, A, F)$, a contradiction.

In the remaining case, i.e. that of a generalization inference, we must have $S \models p$, where F has the form $(\forall x \mid p)$, for some predicate variable x. Since the statements

in S have no free variables we have $\mathsf{Val}(I, A', G) = 1$ for every statement G in S and every assignment A' which is identical to A except on the variable x, so that $\mathsf{Val}(I, A', p) = 1$. But then

$$\mathsf{Val}\big(I, A, \text{`}(\forall x \mid p)\text{'}\big)$$

is the minimum of $\mathsf{Val}(I, A', p)$, taken over all such A', and therefore it follows that $\mathsf{Val}(I, A, \text{`}(\forall x \mid p)\text{'}) = 1$, i.e. $\mathsf{Val}(I, A, F) = 1$, which is again a contradiction. This shows that $S \vdash F$ implies $S \models F$, completing our proof that the conditions $S \models F$ and $S \vdash F$ are equivalent, at least in the case in which S is finite. We will see later that the condition that the set S is finite can be dropped. In fact, we can notice right away that the derivation given above of $S \models F$ from $S \vdash F$ holds also in the case in which S is infinite. Thus, in order to fully establish the generalization of the Gödel completeness theorem, we are only left with proving that $S \models F$ implies $S \vdash F$, for every infinite set S of predicate formulae none of which has occurrences of free variables.

We conclude this subsection by noting that the result just stated fails if the formulae in S are allowed to contain free variables. To see this, consider the simple case in which S consists of the single formula $P(x)$. If this formula were added to the set of axioms of the predicate calculus, we could give the proof

$P(x)$	[axiom]
$\big(\forall x \mid P(x)\big)$	[generalization]
$\big(\forall x \mid P(x)\big) \to P(y)$	[predicate axiom (v)]
$P(y)$	[modus ponens]

Hence we could have $\{P(x)\} \vdash P(y)$. But $\{P(x)\} \models P(y)$ is false, since we can set up a 2-point universe $\mathscr{U} = \{a, b\}$, the assignment $A(x) = a$, $A(y) = b$, and the interpretation I such that $I(P)(a) = 1$ and $I(P)(b) = 0$.

2.2.3 *Working with Universally Valid Predicate Formulae. A Few Simple Examples of Predicate Proof*

A few basic theorems of predicate calculus are needed for later use. One such is

$$\big((\forall x \mid P \to Q)\ \&\ (\exists x \mid P)\big) \to (\exists x \mid Q).$$

The following proof of this statement, and two other sample proofs given later in this section, illustrate some of the techniques of direct, fully detailed predicate proof. By predicate axiom (v) we have

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

and from this by purely propositional reasoning we have

$$(\forall x \mid P \to Q) \to \big((\neg Q) \to (\neg P)\big).$$

By the (strong) rule of generalization this gives

$$(\forall x \mid P \to Q) \to \big(\forall x \mid \big((\neg Q) \to (\neg P)\big)\big).$$

Axiom (ii) now tells us that

$$\big(\big(\forall x \mid \big((\neg Q) \to (\neg P)\big)\big) \,\&\, \big(\forall x \mid (\neg Q)\big)\big) \to \big(\forall x \mid (\neg P)\big),$$

so by propositional reasoning we have

$$(\forall x \mid P \to Q) \to \big(\big(\forall x \mid (\neg Q)\big) \to \big(\forall x \mid (\neg P)\big)\big),$$

and also

$$(\forall x \mid P \to Q) \to \big(\big(\neg\big(\forall x \mid (\neg P)\big)\big) \to \big(\neg\big(\forall x \mid (\neg Q)\big)\big)\big).$$

Since by predicate axiom (iii) we have

$$\big(\neg\big(\forall x \mid (\neg P)\big)\big) \leftrightarrow (\exists x \mid P)$$

and

$$\big(\neg\big(\forall x \mid (\neg Q)\big)\big) \leftrightarrow (\exists x \mid Q),$$

our target statement

$$\big((\forall x \mid P \to Q) \,\&\, (\exists x \mid P)\big) \to (\exists x \mid Q)$$

now follows propositionally.

The following is a useful general principle of the predicate calculus whose universal validity is readily understood intuitively, and which can also be proved formally within the predicate calculus.

Suppose that a predicate formula of the form

$$A \leftrightarrow B$$

has been proved and that F is a syntactically legal predicate formula such that A appears as a subformula of F. Let G be the result of replacing some such occurrence of A in F by an occurrence of B. Then $F \leftrightarrow G$ is also a theorem.

To show this, note that F can be built up starting from A by steps, each of which either joins subformulae together using a propositional operator, or quantifies a formula. Hence it is enough to show that if

$$H_2 \leftrightarrow H_3 \tag{2.5}$$

has already been proved, then

(a) $(H_1 \,\&\, H_2) \leftrightarrow (H_1 \,\&\, H_3)$
(b) $(H_1 \vee H_2) \leftrightarrow (H_1 \vee H_3)$
(c) $(H_1 \leftrightarrow H_2) \leftrightarrow (H_1 \leftrightarrow H_3)$
(d) $(H_1 \rightarrow H_2) \leftrightarrow (H_1 \rightarrow H_3)$
(e) $(H_2 \rightarrow H_1) \leftrightarrow (H_3 \rightarrow H_1)$
(f) $(\neg H_2) \leftrightarrow (\neg H_3)$
(g) $(\forall x \mid H_2) \leftrightarrow (\forall x \mid H_3)$
(h) $(\exists x \mid H_2) \leftrightarrow (\exists x \mid H_3)$

can be proved as well. Notice that (a)–(f) follow readily from (2.5) by propositional reasoning. So to prove our claim we have only to establish that (g) and (h) follow from (2.5) too. This can be shown as follows. By propositional reasoning and the predicate rule of generalization, statement (2.5) yields

$$(\forall x \mid H_2 \rightarrow H_3).$$

By axiom (ii) we have

$$\big((\forall x \mid H_2 \rightarrow H_3) \,\&\, (\forall x \mid H_2)\big) \rightarrow (\forall x \mid H_3),$$

so by propositional reasoning we get

$$(\forall x \mid H_2) \rightarrow (\forall x \mid H_3).$$

The formula

$$(\forall x \mid H_3) \rightarrow (\forall x \mid H_2)$$

can be derived in the same way, and so we have

$$(\forall x \mid H_2) \leftrightarrow (\forall x \mid H_3).$$

Since (2.5) yields

$$(\neg H_2) \leftrightarrow (\neg H_3)$$

by propositional reasoning, it follows in the same way that

$$\big(\forall x \mid (\neg H_2)\big) \leftrightarrow \big(\forall x \mid (\neg H_3)\big)$$

and so

$$\big(\neg\big(\forall x \mid (\neg H_2)\big)\big) \leftrightarrow \big(\neg\big(\forall x \mid (\neg H_3)\big)\big).$$

It follows by predicate axiom (iii) and propositional reasoning that

$$(\exists x \mid H_2) \leftrightarrow (\exists x \mid H_3),$$

completing the proof of our claim.

The following ‘change of bound variables’ law is still another rule of obvious universal validity, which as usual can be proved formally within the predicate calculus.

Let F be a syntactically well-formed predicate formula containing x as a free variable, let y be a variable not occurring in F, and let $F(x \hookrightarrow y)$ be the result of replacing every free occurrence of x by an occurrence of y. Then

$$(\forall x \mid F) \leftrightarrow \big(\forall y \mid F(x \hookrightarrow y)\big)$$

and

$$(\exists x \mid F) \leftrightarrow \big(\exists y \mid F(x \hookrightarrow y)\big)$$

are universally valid predicate formulae. To show this, we first use predicate axiom (v) to get

$$(\forall x \mid F) \rightarrow F(x \hookrightarrow y),$$

and so

$$(\forall x \mid F) \rightarrow \big(\forall y \mid F(x \hookrightarrow y)\big)$$

follows by the (strong) rule of generalization, since y does not occur freely in $(\forall x \mid F)$.

Since replacing each free occurrence of x in F by y and then each y by x brings us back to the original x, we have

$$F(x \hookrightarrow y)(y \hookrightarrow x) = F.$$

Thus the argument just given can be used again to show that

$$\big(\forall y \mid F(x \hookrightarrow y)\big) \rightarrow (\forall x \mid F),$$

and so it results propositionally that

$$\big(\forall y \mid F(x \hookrightarrow y)\big) \leftrightarrow (\forall x \mid F).$$

Applying the same argument to '$\neg F$' we can get

$$\big(\neg\big(\forall y \mid \neg F(x \hookrightarrow y)\big)\big) \leftrightarrow \big(\neg(\forall x \mid \neg F)\big),$$

and so

$$\big(\exists y \mid F(x \hookrightarrow y)\big) \leftrightarrow (\exists x \mid F),$$

using predicate axiom (iii).

The observations just made allow any predicate formula F to be transformed, via a sequence of formulae all provably equivalent to each other, into an equivalent formula G all of whose quantifiers appear to the extreme left of the formula. To achieve this, we must also use the following auxiliary group of predicate rules, which apply if the variable x does not occur freely in Q:

(a) $(\forall x \mid P \vee Q) \leftrightarrow ((\forall x \mid P) \vee Q)$
(b) $(\forall x \mid P \,\&\, Q) \leftrightarrow ((\forall x \mid P) \,\&\, Q)$

(c) $(\forall x \mid P \to Q) \leftrightarrow ((\exists x \mid P) \to Q)$
(d) $(\forall x \mid Q \to P) \leftrightarrow (Q \to (\forall x \mid P))$
(e) $(\exists x \mid P \vee Q) \leftrightarrow ((\exists x \mid P) \vee Q)$
(f) $(\exists x \mid P \,\&\, Q) \leftrightarrow ((\exists x \mid P) \,\&\, Q)$
(g) $(\exists x \mid P \to Q) \leftrightarrow ((\forall x \mid P) \to Q)$
(h) $(\exists x \mid Q \to P) \leftrightarrow (Q \to (\exists x \mid P))$.

These rules can be proved as follows. Predicate axiom (v) gives

$$(\forall x \mid P) \to P,$$

and so by propositional reasoning from the tautology

$$\big((\forall x \mid P) \to P\big) \to \big(\big((\forall x \mid P) \vee Q\big) \to (P \vee Q)\big),$$

we get

$$\big((\forall x \mid P) \vee Q\big) \to (P \vee Q).$$

Since x does not occur freely in $((\forall x \mid P) \vee Q)$, generalization now gives

$$\big((\forall x \mid P) \vee Q\big) \to (\forall x \mid P \vee Q).$$

Conversely we get

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

from predicate axiom (v), and so

$$\big((\forall x \mid P \vee Q) \,\&\, (\neg Q)\big) \to P.$$

Since x does not occur freely in $((\forall x \mid P \vee Q) \,\&\, (\neg Q))$, by generalization we get

$$\big((\forall x \mid P \vee Q) \,\&\, (\neg Q)\big) \to (\forall x \mid P),$$

and then

$$(\forall x \mid P \vee Q) \to \big((\forall x \mid P) \vee Q\big),$$

so altogether

$$(\forall x \mid P \vee Q) \leftrightarrow \big((\forall x \mid P) \vee Q\big),$$

proving (a).

To prove (b) we reason as follows.

$$(\forall x \mid P \,\&\, Q) \to (P \,\&\, Q)$$

by axiom (v), so

$$(\forall x \mid P \,\&\, Q) \to P$$

by propositional reasoning. Since x does not occur freely in $(\forall x \mid P \,\&\, Q)$, by generalization we derive

$$(\forall x \mid P \,\&\, Q) \to (\forall x \mid P)$$

from this. Thus, by propositional reasoning, we obtain

$$(\forall x \mid P \,\&\, Q) \to \big((\forall x \mid P) \,\&\, Q\big).$$

Conversely, since

$$\big((\forall x \mid P) \,\&\, Q\big) \to (\forall x \mid P)$$

we have

$$\big((\forall x \mid P) \,\&\, Q\big) \to P$$

by axiom (v) and propositional reasoning. Since

$$\big((\forall x \mid P) \,\&\, Q\big) \to Q$$

is propositional, we get

$$\big((\forall x \mid P) \,\&\, Q\big) \to (P \,\&\, Q),$$

and now

$$\big((\forall x \mid P) \,\&\, Q\big) \to (\forall x \mid P \,\&\, Q)$$

follows by generalization, since x does not occur freely in $(\forall x \mid P) \,\&\, Q$. Altogether this gives

$$\big((\forall x \mid P) \,\&\, Q\big) \leftrightarrow (\forall x \mid P \,\&\, Q),$$

i.e. (b).

Statement (c) now follows via the chain of equivalences

$$\begin{aligned}(\forall x \mid P \to Q) &\leftrightarrow \big(\forall x \mid (\neg P) \vee Q\big)\\ &\leftrightarrow \big(\big(\forall x \mid (\neg P)\big) \vee Q\big)\\ &\leftrightarrow \big(\big(\neg\big(\forall x \mid (\neg P)\big)\big) \to Q\big)\\ &\leftrightarrow \big((\exists x \mid P) \to Q\big).\end{aligned}$$

Similarly statement (d) follows via the chain of equivalences

$$\begin{aligned}(\forall x \mid Q \to P) &\leftrightarrow \big(\forall x \mid (\neg Q) \vee P\big)\\ &\leftrightarrow \big((\neg Q) \vee \big(\forall x \mid P\big)\big)\\ &\leftrightarrow \big(Q \to \big(\forall x \mid P\big)\big).\end{aligned}$$

The proofs of (e–h) are left to the reader.

2.2.4 The Prenex Normal Form of Predicate Formulae

The prenex normal form of a predicate formula F is a logically equivalent formula in which quantifiers $\forall$ and $\exists$ appear only at the very start of the formula. Rules (a–h) can now be used iteratively in the following way to put an arbitrary formula F into prenex normal form. We first change bound variables, using the equivalences derived above for this purpose, to ensure that all bound variables are distinct and that no bound variable is the same as any variable occurring freely. Then we use equivalences

$$(P \leftrightarrow Q) \leftrightarrow \big((P \rightarrow Q)\ \&\ (Q \rightarrow P)\big)$$

to replace all '$\leftrightarrow$' operators in our formula with combinations of implication and conjunction operators. After this, we search the syntax tree of the formula, looking for all quantifier nodes whose parent nodes are not already quantifier nodes, and moving them upward in a manner to be described. If there are no such nodes, then all the quantifiers occur in an unbroken sequence starting at the tree root, and so in the unparsed form of the formula they all occur at the left of the formula. The quantifier node moved at any moment should always be one that is as close as possible to the root of the syntax tree. Given that the parent of this quantifier is not itself a quantifier node, the parent must be marked with one of the Boolean operators &, $\vee$, $\rightarrow$, $\neg$. If the operator at the parent node is '$\neg$', we use one of the equivalences

$$(\forall x_1, \ldots, x_k \mid \neg P) \leftrightarrow \big(\neg(\exists x_1, \ldots, x_k \mid P)\big)$$

and

$$(\exists x_1, \ldots, x_k \mid \neg P) \leftrightarrow \big(\neg(\forall x_1, \ldots, x_k \mid P)\big)$$

to interchange the positions of the '$\neg$' operator and the quantifier. In the remaining cases we use one of the equivalences (a–h) to achieve a like interchange. When this process, each of whose steps transforms our original formula into an equivalent formula, can no longer continue, the formula that remains will clearly be in prenex normal form.

2.2.5 The Deduction Theorem

The Deduction Theorem of predicate calculus, which will be useful below, states that (provided that neither F or any of the statements in S contain any free variables) the implication $F \rightarrow G$ can be proved from a set S of predicate axioms if and only if G can be proved if F is added to the set S of axioms. Note that this is an easy consequence of the Gödel Completeness Theorem in the generalized form discussed at the start of this section. But in what follows we need to know that this result can be proved directly. This will now be shown.

Theorem 2.2 (Deduction) *Let S be a collection of predicate formulae with no free variables and let S' be obtained from S by adding to it a predicate formula F with no free variables. Then*

$$S \vdash F \to G \quad \textit{if and only if} \quad S' \vdash G,$$

for any predicate formula G.

Proof Let S, S', F, and G be as above. First assume that $S \vdash F \to G$ holds and let

$$H_1, H_2, \ldots, H_n,$$

with $H_n = F \to G$, be a proof of $F \to G$ from S. Then it follows immediately that

$$H_1, H_2, \ldots, H_n, F, G$$

is a proof of G from S'.

Conversely, assume that $S' \vdash G$ and let

$$H_1, H_2, \ldots, H_n, \tag{2.6}$$

with $H_n = G$, be a proof of G from S'. We can suppose without loss of generality that this proof does not use the strong variant of the rule of generalization stated earlier, but only the weaker form of this rule. Consider the sequence of predicate formulae

$$F \to H_1, \quad F \to H_2, \quad \ldots, \quad F \to H_n. \tag{2.7}$$

We will show that by inserting suitable auxiliary formulae into this sequence we can turn it into a proof from S of $F \to G$. Indeed, for each $i = 1, 2, \ldots, n$ one of the following cases will apply:

(i) H_i may be a predicate axiom or H_i may be an element of S. In this case we insert the formulae

$$H_i$$
$$H_i \to (F \to H_i)$$

(of which the latter is a tautology) into (2.7) just before the formula $F \to H_i$.

(ii) H_i may follow from H_j and $H_k = H_j \to H_i$ by modus ponens step. In this case we insert the formulae

$$(F \to H_j) \to \big(\big(F \to (H_j \to H_i)\big) \to (F \to H_i)\big)$$
$$\big(F \to (H_j \to H_i)\big) \to (F \to H_i)$$

(of which the former is a tautology) into (2.7) just before the formula $F \to H_i$.

(iii) In the remaining possible cases, namely if H_i is derived from some earlier statement of (2.6) by the rule of generalization, or if $H_i = F$, we need not add any formula to (2.7).

Let

$$K_1, K_2, \ldots, K_m$$

be the sequence of predicate formulae generated in the manner just described. It is easy to check that this sequence constitutes a proof of $K_m = F \to G$ from S, provided that we now allow use of the strong variant of the rule of generalization. Since, as shown above, any such proof can be transformed into one in which all uses of the strong variant of the rule of generalization have been eliminated and only the weak form of this rule is used, it follows that $S \vdash F \to G$, concluding our proof of the deduction theorem. □

The deduction theorem admits the following semantic version, whose proof is left to the reader.

Theorem 2.3 *Let S, S', F, and G be as in the statement of the deduction theorem. Then*

$$S \models F \to G \quad \textit{if and only if} \quad S' \models G.$$

2.2.6 *Definitions in Predicate Calculus; the Notion of 'Conservative Extension'*

Since the use of definitions to introduce new predicate and function symbols is fundamental to ordinary mathematical practice, it is important to understand the sense in which the predicate calculus accommodates this notion. The simplest definitions are algebraic, i.e. they simply introduce names for compound expressions written in terms of previously defined predicate and function symbols. Such definitions are unproblematical, since any use of them can be eliminated by expanding the new name back into the underlying expression which it abbreviates. But another, less trivial kind of definition is also essential. This is known as *definition by introduction of Skolem functions*. More specifically, once we have proved a formula of the form

$$\big(\forall y_1, \ldots, y_n \,\big|\, \big(\exists z \mid P(y_1, \ldots, y_n, z)\big)\big) \tag{2.8}$$

using the axioms of predicate calculus and some set S of additional axioms (none of which should have any free variables), we can introduce any desired new, never previously used function name f and add the statement

$$\big(\forall y_1, \ldots, y_n \mid P\big(y_1, \ldots, y_n, f(y_1, \ldots, y_n)\big)\big) \tag{2.9}$$

to S. The point is that, although this added statement clearly allows us to prove new statements concerning the newly introduced symbol f, it does not make it possible to prove any statement *not involving* f that could not have been proved without its introduction.

This very important result can be called the *fundamental principle of definition*. To prove it we argue as follows. (But note that the following proof uses the Gödel Completeness Theorem, and so is entirely nonconstructive, i.e. it does not tell us how to produce the definition-free proof whose existence it asserts.) Let P, S, and f be as above, and let S' be obtained from S by adjoining the formula (2.9) to S. Let F be a formula not involving the symbol f, and suppose that $S' \vdash F$. Then we have $S' \models F$ by the Gödel completeness theorem (as extended above). Our goal is to show that $S \vdash F$. By the Gödel completeness theorem it is enough to show that $S \models F$. To this purpose, let $(\mathscr{U}, I, A)$ be an interpretation framework covering F and the statements in S and such that $\mathsf{Val}(I, A, G) = 1$ for each G in S. Then we must show that $\mathsf{Val}(I, A, F) = 1$.

Introduce an auxiliary Boolean function $p(u_1, \ldots, u_n, u_{n+1})$, mapping the Cartesian product $\mathscr{U}^{n+1}$ of $n+1$ copies of $\mathscr{U}$ into $\{0, 1\}$, by setting

$$p(u_1, \ldots, u_n, u_{n+1}) = \mathsf{Val}\big(I, A(u_1, \ldots, u_n, u_{n+1}),\ \text{`}P(y_1, \ldots, y_n, z)\text{'}\big),$$

where $A(u_1, \ldots, u_n, u_{n+1})$ is the assignment which agrees with A everywhere except on the variables $y_1, \ldots, y_n$ and z, for which variables we take

$$\begin{aligned} A(u_1, \ldots, u_n, u_{n+1})(y_1) &= u_1, \\ \vdots \qquad\qquad & \vdots\ \vdots \\ A(u_1, \ldots, u_n, u_{n+1})(y_n) &= u_n, \\ A(u_1, \ldots, u_n, u_{n+1})(z) \ &= u_{n+1}. \end{aligned}$$

Since

$$S \vdash \big(\forall y_1, \ldots, y_n \,\big|\, \big(\exists z \mid P(y_1, \ldots, y_n, z)\big)\big),$$

we have

$$S \models \big(\forall y_1, \ldots, y_n \,\big|\, \big(\exists z \mid P(y_1, \ldots, y_n, z)\big)\big)$$

and therefore

$$\begin{aligned} 1 &= \mathsf{Val}\big(I, A, \big(\forall y_1, \ldots, y_n \,\big|\, \big(\exists z \mid P(y_1, \ldots, y_n, z)\big)\big)\big) \\ &= \min_{u_1, \ldots, u_n}\big(\max_{u_{n+1}}\big(\mathsf{Val}\big(I, A(u_1, \ldots, u_n, u_{n+1}),\ P(y_1, \ldots, y_n, z)\big)\big)\big) \\ &= \min_{u_1, \ldots, u_n}\big(\max_{u_{n+1}}\big(p(u_1, \ldots, u_n, u_{n+1})\big)\big), \end{aligned}$$

where the minima and maxima over the subscripts seen extend over all values in $\mathscr{U}$. Hence there exists a function h from $\mathscr{U}^n$ into $\mathscr{U}$ such that

$$p\big(u_1, \ldots, u_n, h(u_1, \ldots, u_n)\big) = 1$$

for all $u_1, \ldots, u_n$ in $\mathscr{U}$. Let I' be an interpretation which agrees with I everywhere except on the function symbol f and such that $I'(f)$ is the function h just defined

(which is, as required, a mapping from $\mathscr{U}^n$ to $\mathscr{U}$). Hence

$$\begin{aligned} 1 &= \min_{u_1,\ldots,u_n}\big(p\big(u_1,\ldots,u_n,h(u_1,\ldots,u_n)\big)\big) \\ &= \min_{u_1,\ldots,u_n}\big(\mathsf{Val}\big(I', A(u_1,\ldots,u_n),\ P\big(y_1,\ldots,y_n,f(y_1,\ldots,y_n)\big)\big)\big) \\ &= \mathsf{Val}\big(I', A,\ \big(\forall y_1,\ldots,y_n \mid P\big(y_1,\ldots,y_n,f(y_1,\ldots,y_n)\big)\big)\big), \end{aligned}$$

where $A(u_1,\ldots,u_n)$ is the assignment which agrees with A everywhere except on the variables $y_1,\ldots,y_n$, for which variables we take

$$\begin{aligned} A(u_1,\ldots,u_n)(y_1) &= u_1, \\ \vdots \qquad &\quad \vdots \\ A(u_1,\ldots,u_n)(y_n) &= u_n. \end{aligned}$$

Since no formula G in S involves the function symbol f, we have

$$\mathsf{Val}(I', A, G) = \mathsf{Val}(I, A, G) = 1, \quad \text{for all } G \text{ in } S.$$

Therefore

$$\mathsf{Val}(I', A, F) = 1,$$

since, as observed above, $S' \models F$. But since the formula F does not involve the function symbol f, we have

$$\mathsf{Val}(I, A, F) = 1,$$

proving that $S \models F$, and so $S \vdash F$. This concludes our proof of the fundamental principle of definition.

The central notion implicit in the preceding argument is worth capturing formally.

Definition 2.8 Let S be a set of predicate formulae not involving any free variables, and let S' be a larger such set (possibly involving function and predicate symbols that do not occur in S). Then S' is called a *conservative extension* of S if

$$S' \vdash F \quad \text{implies} \quad S \vdash F,$$

for every formula F involving no predicate or function symbols not present in one of the formulae of S.

The argument just given shows that the addition of formula (2.9) to any set S of formulae not containing free variables for which (2.8) can be proved yields a conservative extension.

2.2.7 *Proof of the Gödel Completeness Theorem*

Now we come to the proof of the Gödel completeness theorem. To prove it we first show, without using it, that the theorem holds for a certain very limited form of Skolem definition, namely if we introduce a single new constant symbol C (i.e. function symbol of 0 arguments) satisfying $P(C)$, provided that we have previously proved a predicate formula of the form

$$\big(\exists z \mid P(z)\big).$$

These constants are traditionally called *Henkin constants*, after Leon Henkin, who introduced the technique that we will use. Our first key lemma is as follows.

Lemma 2.1 *Let S be a collection of (syntactically well-formed) predicate formulae without free variables and let C be a constant symbol not appearing in any of the formulae of S. For each formula H, let $H(C \hookrightarrow x)$ denote the result of replacing each occurrence of C in H by an occurrence of x, where x designates a variable not otherwise used. Then, if $S \vdash H$, we have*

$$S \vdash H(C \hookrightarrow x).$$

In intuitive terms, this lemma tells us that if the axioms S can be used to prove some statement about a constant which they never mention, they can be used to prove the same statement in which C is replaced by a variable.

Proof Suppose that Lemma 2.1 fails for some H. Then, proceeding inductively, we can suppose that Lemma 2.1 holds for all statements having proofs shorter than that of H. Without loss of generality, we can assume that the variable x is not used in the proof of H. Consider the final step in the proof of H. This must either be (i) a citation of a predicate axiom; (ii) a citation of some statement in S; (iii) a modus ponens step involving two formulae G and $G \to H$ proved earlier; (iv) a generalization step from a formula G proved earlier. Concerning case (i), if H is a predicate axiom so is $H(C \hookrightarrow x)$. In case (ii), namely if H is a member of S, H cannot involve the constant C, so that $H(C \hookrightarrow x) = H$ and therefore we plainly have $S \vdash H(C \hookrightarrow x)$.

Next consider case (iii). Since in this case G and $G \to H$ both have shorter proofs than that of H, it follows by inductive assumption that $S \vdash G(C \hookrightarrow x)$ and $S \vdash (G \to H)(C \hookrightarrow x)$, i.e. $S \vdash G(C \hookrightarrow x) \to H(C \hookrightarrow x)$. Therefore it follows by a modus ponens step that $S \vdash H(C \hookrightarrow x)$.

Finally we consider case (iv). In this case G has a shorter proof than that of its generalization $H = (\forall z \mid G)$. Hence by inductive assumption $S \vdash G(C \hookrightarrow x)$, so that, by the rule of generalization, $S \vdash (\forall z \mid G(C \hookrightarrow x))$ and therefore $S \vdash H(C \hookrightarrow x)$, since

$$H(C \hookrightarrow x) = (\forall z \mid G)(C \hookrightarrow x) = \big(\forall z \mid G(C \hookrightarrow x)\big),$$

proving our claim in case (iv) and thus completing our proof of Lemma 2.1. □

Next we prove the following consequence of Lemma 2.1.

Lemma 2.2 *Let S be a collection of (syntactically well-formed) predicate formulae without free variables. Let F be a predicate formula involving the one free variable y. Let C be a constant symbol not appearing in any of the formulae of S or in F, and let $F(y \hookrightarrow C)$ denote the formula obtained from F by replacing each occurrence of y by an occurrence of C. Suppose that*

$$S \vdash (\exists y \mid F).$$

Let S' be the union of S and the statement $F(y \hookrightarrow C)$. Then S' is a conservative extension of S.

Proof Let H be a formula involving only the symbols appearing in S, so that in particular the constant C does not occur in H. Suppose that $S' \vdash H$. By the Deduction Theorem we have

$$S \vdash F(y \hookrightarrow C) \to H.$$

By Lemma 2.1 this last formula yields

$$S \vdash \big(F(y \hookrightarrow C) \to H\big)(C \hookrightarrow x),$$

where x is a variable not otherwise used. Therefore

$$S \vdash F(y \hookrightarrow x) \to H,$$

since $F(y \hookrightarrow C)(C \hookrightarrow x) = F(y \hookrightarrow x)$ and $H(C \hookrightarrow x) = H$. Applying the rule of generalization we obtain

$$S \vdash \big(\forall x \mid F(y \hookrightarrow x) \to H\big).$$

We have shown above that

$$\Big(\big(\forall x \mid F(y \hookrightarrow x) \to H\big) \,\&\, \big(\exists x \mid F(y \hookrightarrow x)\big)\Big) \to (\exists x \mid H)$$

and

$$(\exists y \mid F) \leftrightarrow \big(\exists x \mid F(y \hookrightarrow x)\big)$$

are universally valid. Thus, by propositional reasoning,

$$S \vdash (\exists x \mid H).$$

But since the variable x does not occur freely in H, we have

$$\vdash \big(\forall x \mid (\neg H)\big) \leftrightarrow (\neg H)$$

by predicate axiom (iv), and so it follows propositionally that

$$\vdash \neg\big(\forall x \mid (\neg H)\big) \leftrightarrow H.$$

Predicate axiom (iii) then gives

$$\vdash (\exists x \mid H) \leftrightarrow H$$

and so $S \vdash H$, proving that S' is a conservative extension of S. □

2.2.7.1 The Remainder of the Proof: Predicate Consistency Principle

We will now complete our proof of the Gödel completeness theorem. For this, it is convenient to restate it in the following way.

Predicate consistency principle *Let S be a set of formulae, none containing free variables, such that S is* consistent*, i.e. $S \vdash \mathsf{false}$ is false. Then there exists a* model *for S, i.e. an interpretation framework $(\mathscr{U}, I, A)$ covering all the predicate and function symbols appearing in S, such that $\mathsf{Val}(I, A, F) = 1$ for each F in S. Conversely if there is a model for S then S is consistent.*

This is simply the statement that $S \vdash \mathsf{false}$ is false iff $S \models \mathsf{false}$ is false. For '$S \models \mathsf{false}$ is false' means that there is an interpretation framework $(\mathscr{U}, I, A)$ covering all the statements F in S such that $\mathsf{Val}(I, A, F) = 1$ for each F in S, but nonetheless satisfying the (required) condition that $\mathsf{Val}(I, A, \mathsf{false}) = 0$.

It is an easy matter to see that the predicate consistency principle implies that for every set S of predicate formulae with no free variables and for every predicate formula F the following condition holds:

$$\textbf{if } S \models F \textbf{ then } S \vdash F. \tag{2.10}$$

Indeed, assume that $S \models F$ holds and that $S \vdash F$ is false. Then $S \vdash (\forall v_1, \ldots, v_n \mid F)$, where $v_1, \ldots, v_n$ are the free variables of F, must also be false, because otherwise by repeated use of axiom (v) and the rule of modus ponens $S \vdash F$ would follow. Let S' be the set of predicate formulae obtained by adding the formula $\neg(\forall v_1, \ldots, v_n \mid F)$ to S. Then $S' \vdash \mathsf{false}$ must be false, because otherwise by the deduction theorem

$$S \vdash \neg(\forall v_1, \ldots, v_n \mid F) \rightarrow \mathsf{false}$$

would hold and therefore, by propositional reasoning, $S \vdash (\forall v_1, \ldots, v_n \mid F)$ would hold. Therefore the predicate consistency principle implies that S' has a model, namely there exists an interpretation framework $(\mathscr{U}, I, A)$ covering all the statements G of S' and such that $\mathsf{Val}(I, A, G) = 1$ for all such G. Thus, in particular, we have $\mathsf{Val}(I, A, C) = 1$ for all the formulae C in S and $\mathsf{Val}(I, A, \neg(\forall v_1, \ldots, v_n \mid F)) = 1$. This last statement implies that there exists an assignment A' such that $\mathsf{Val}(I, A', F) = 0$. Since all formulae in S have no free variables, it follows that $\mathsf{Val}(I, A', C) = \mathsf{Val}(I, A, C) = 1$ for each formula C in S, thus contradicting our initial assumption that $S \models F$ holds, and thereby proving statement (2.10).

But the statement (2.10) implies, and indeed is a bit more general than, the Gödel completeness theorem. This shows that the Gödel completeness theorem will follow if we can prove the predicate consistency principle.

Proof To this end assume first that S is not consistent. Then $S \vdash \mathsf{false}$ holds. But then, as was shown earlier, $S \models \mathsf{false}$ follows, so that S cannot have any model.

For the converse, assume that S is consistent, in which case we must show that S has a model. We can and shall suppose that all our formulae are in prenex normal form, since we have seen that given any set of formulae there is an equivalent set of prenex normal formulae. We proceed in a kind of 'algorithmic' style, to generate a steadily increasing collection of formulae known to be consistent. At the end of this process it will be easy to construct a model of the set S of statements using these formulae and a bit of purely propositional reasoning. The idea of the proof is to introduce enough new constants C to ensure that, for each original existentially quantified formula

$$(\exists x \mid F),$$

there exists a C for which

$$F(x \hookrightarrow C)$$

is known to be true. To this end, we maintain the following lists and sets of formulae, along with one set of auxiliary constants. These lists and sets can be (countably) infinite and will steadily grow larger. In order to be certain that there exist only finitely many constants with names below any given length, it will be convenient for us to suppose that all constants have names like 'C', 'CC', 'CCC', ... The lists and sets we maintain are then:

SC: the set of all constants introduced so far.
SUF: the set of all universally quantified formulae generated so far.
SNQ: the set of all formulae containing no quantifiers generated so far.
LEF: the list of all existentially quantified formulae generated so far.
This list is always kept in order of increasing length of the formulae on it. Formulae of the same length are arranged in alphabetical order. Each formula on the list LEF is marked either as 'processed' or 'unprocessed'.

These data objects are initialized as follows. SC initially contains all the constants appearing in functions of S. SUF contains all the formulae of S which start with a universal quantifier. SNQ contains all the formulae of S which contain no quantifiers. LEF contains all the formulae of S which start with an existential quantifier. These are arranged in the order just described. All the formulae on LEF are originally marked 'unprocessed'.

The auxiliary set FS consists of all function symbols appearing in formulae of S.

The following processing steps are repeated as often as they apply, causing our four data objects to grow steadily. Note that SC is always finite, becoming infinite only in the limit, but that SUF, SNQ, and LEF can be infinite during the process that we now describe.

(a) Whenever new constants are added to SC or new universally quantified formulae to SUF, all the constants on SC are combined in all possible ways with function symbols of FS to create new terms, and these terms are substituted in all possible ways for initial universally quantified variables in formulae of SUF (all the variables up to the first existentially quantified variable, if any), thereby generating new formulae, some starting with existential quantifiers (these are added to LEF if not already there, following which LEF is rearranged into its required order), others with no quantifiers at all (these are added to SNQ if not already there).

(b) After each step (a), or if no step (a) is needed, we examine LEF to find the first formula $(\exists x \mid F)$ on it not yet marked 'processed'. For this formula, we generate a new constant symbol C, build the formula $F(x \hookrightarrow C)$ produced by replacing each free occurrence of x in F by C, and add this formula to SUF or LEF or SNQ, depending on whether it starts with a universal quantifier, starts with an existential quantifier, or has no quantifiers at all, and finally add the new constant C to SC. It is understood that the list LEF must always be maintained in lexicographic order. Finally, the formula $(\exists x \mid F)$ on LEF is then marked 'processed'.

Processing begins as if the set of constants appearing in the formulae of S have just been added to SC, and so with step (a). (If there are no such constants, we must generate one initial constant symbol C to start processing.)

At the end of this (perhaps infinitely long) sequence of processing steps, we may have generated a countably infinite list of constants as SC, and put infinitely many formulae into both of the sets SUF and SNQ and on the list LEF. But we can be sure that it is never possible to prove a contradiction from our set of formulae. For otherwise a contradiction would result from some finite set of formulae, all of which would have been added to our collection at some stage in the process we have described. But by assumption our formulae are consistent to begin with. Moreover no step of type (a) can spoil consistency, since only predicate consequences of previously added formulae are added during such steps. Nor can steps of type (b) spoil consistency, since it was proved above that steps of this kind yield conservative extensions of the set of formulae previously present.

It follows that at the end of the process we have described the set SNQ of unquantified formulae that results is consistent, i.e. that every finite subset of this set of formulae is consistent. We have proved above that this implies that SNQ has a propositional model, i.e. that we can assign a $0/1$ value $\mathsf{Va}(T)$ to each atomic formula T appearing in any of the formulae F of SNQ, in such a way that each such F evaluates to 'true' if the atomic formulae appearing in it are replaced by these values, and the standard rules for calculating Boolean truth values of propositional combinations are then applied. Note for use below that each of the atomic formulae T of the set AT of all such formulae appearing in any F has the form $P(t_1, \ldots, t_k)$, where P is a predicate symbol and $t_1, \ldots, t_k$ are 'constant' terms (i.e. terms devoid of variables).

Now we show that there exists a model whose universe is the set CT of all constant terms generated by applying the function symbols in FS to the constants in

SC in all possible ways. (The resulting set of terms is the so called *free universe* FU generated by these constants and the function symbols in FS.) Each k-adic function symbol f in FS is trivially associated with a mapping $I(f)$ from the Cartesian product FU^k of k copies of FU into FU, namely we can put

$$I(f)(t_1, \ldots, t_k) = f(t_1, \ldots, t_k)$$

for all lists $t_1, \ldots, t_k$ of terms. For this I and every possible assignment A it is immediate that

$$\mathsf{Val}(I, A, t) = t$$

for each term t in FU. A 0/1 valued function on FU^k can now be associated with each predicate symbol P appearing in a formula of S, namely we can write

$$I(P)(t_1, \ldots, t_k) = \mathsf{Va}\big(P(t_1, \ldots, t_k)\big)$$

for each atomic formula $P(t_1, \ldots, t_k)$ appearing in one of the formulae of SNQ, and define $I(P)(t_1, \ldots, t_k)$ arbitrarily for all other atomic formulae; here 'Va' is the Boolean assignment of truth values described in the preceding paragraph. It is then immediate that for every assignment A we have

$$\mathsf{Val}(I, A, F) = 1,$$

for each formula of SNQ. It remains to be shown that we must have $\mathsf{Val}(I, A, F) = 1$ for the quantified formulae of SUF and LEF also and for every assignment A. Suppose that this is not the case. Then there exists a formula F with $n > 0$ quantifiers for which $\mathsf{Val}(I, A, F) = 0$. Proceeding inductively, we may suppose that n is the smallest number of quantifiers for which this is possible. If F belongs to LEF, then it has the form $(\exists x \mid G)$, and by construction we will have added a formula of the form $G(x \hookrightarrow C)$, with some constant symbol C, to our collection. Since $G(x \hookrightarrow C)$ has fewer quantifiers than n, we must have $\mathsf{Val}(I, A, G(x \hookrightarrow C)) = 1$, and so $\mathsf{Val}(I, A, F)$, which is the maximum over a collection of values including $\mathsf{Val}(I, A, G(x \hookrightarrow C))$, must be 1 also.

It only remains to consider the case in which F belongs to SUF, and so has the form

$$(\forall x_1, \ldots, x_m \mid G)$$

for some G. In this case, all formulae $G(x_1 \hookrightarrow t_1, \ldots, x_m \hookrightarrow t_m)$, where $t_1, \ldots, t_m$ are any terms in our universe, namely the set TERM of all constant terms generated by applying the function symbols in FS to the constants in SC in all possible ways, will have been added to our collection. All these formulae have fewer quantifiers than n, and so we must have

$$\mathsf{Val}\big(I, A, G(x_1 \hookrightarrow t_1, \ldots, x_m \hookrightarrow t_m)\big) = 1$$

for all these terms. Hence the minimum of all these values, namely

$$\mathsf{Val}\big(I, A, (\forall x_1, \ldots, x_m \mid G)\big)$$

must also have the value 1. This completes our proof of the predicate consistency principle and in turn of the Gödel completeness theorem. □

The argument just given clearly leads to the following slightly stronger result.

Corollary 2.1 *Let S be a set of formulae in prenex normal form, and let SNQ be the set of all unquantified formulae generated by the process described above. Then S is consistent, i.e. it has a model, if and only if SNQ, regarded as a collection of propositions whose propositional symbols are the atomic formulae appearing in SNQ, is propositionally consistent.*

Proof As shown above, the set of statements in SNQ must be consistent if S is consistent. The argument given above establishes the converse, i.e. it shows that S has a model if SNQ is propositionally consistent. □

2.2.7.2 Immediate Consequences of the Gödel Completeness Theorem

The preceding corollary implies that in situations in which we can be sure that the procedure described in the proof of the predicate consistency principle will produce sets SC, SUF, SNQ, and a list LEF all of which remain finite, this procedure can be used as an algorithm to decide in a finite number of steps whether or not a given finite set S of prenex normal formulae (none of which involves free variables) is consistent. One case in which this remark applies is that of pure '$\exists \cdots \exists \forall \cdots \forall$' formulae, as defined by the following conditions:

i. S is a finite set of formulae in prenex normal form not involving free variables.
ii. No formula in S involves function symbols of arity greater than zero (i.e., the only terms allowed in these formulae are variables and constant terms). Of course, any number of predicate symbols can be used.
iii. No existential quantifier can follow a universal quantifier in any formula of S.

Note that the condition iii, implies that the sequence of quantifiers prefixed to any '$\exists \cdots \exists \forall \cdots \forall$' formula has the form

$$(\exists y_1, \ldots, y_m \mid (\forall x_1, \ldots, x_n \mid \cdots$$

To see why in this case the procedure described in the proof of the predicate consistency principle must converge after a finite number of steps, note first of all that since there are no function symbol the only terms substituted for universally quantified variables in step (a) of that procedure are constants. These constants must either be present in our initial formulae or be generated in some step of the procedure described. But since all existential quantifiers precede all universal quantifiers, the aforesaid step (a) will never generate any new formula containing existential quantifiers. Hence the number of constants generated is no greater than the number of existential quantifiers contained in our original collection of formulae, and substitution of these for all the universally quantified variables present will generate no more than a finite set of formulae.

Decidability for the Bernays–Schönfinkel Sentences An interesting special case of the foregoing is that when we are given a finite set S of pure '$\exists \cdots \exists \forall \cdots \forall$' formulae, involving no free variables, as described above, and one additional formula F of the same kind and in which no universal quantifier follows an existential quantifier, and we want to determine whether $S \vdash F$ holds. Let S' be the set of formulae obtained by adding the formula '$\neg F$' to S. Then we know that $S \vdash F$ holds if and only S' is inconsistent. But by moving the connective $\neg$ in '$\neg F$' across the quantifier prefix of F, we obtain another set S^* which is equivalent to S' and is still a finite set of pure '$\exists$-$\forall$' formulae, whose consistency can be tested algorithmically in the manner just explained.

The Löwenheim–Skolem Theorem The argument given in the proof of the predicate consistency principle allows us to derive another interesting fact, known as the Löwenheim–Skolem Theorem. This states that any consistent countable set of sentences has a countable model. Indeed, if S is countable (as was implicitly assumed in our proof of the predicate consistency principle) then all the sets SC, SUF, SNQ, FS, and the list LEF maintained by the process described in the proof of the predicate consistency principle are countable at each stage, and so must also be countable in the limit. Therefore the model constructed from SNQ using the technique seen above must also be countable.

The Compactness Theorem A set S of predicate formulae is said to be *satisfiable* if it has a model. The Compactness Theorem states that if S is a set of predicate sentences such that every finite subset of S is satisfiable, then the whole infinite set S is satisfiable. This theorem is an easy consequence of the predicate consistency principle. Indeed, let S be a set of predicate sentences such that every finite subset of S has a model, and assume that S is not satisfiable. Then $S \models \mathsf{false}$ holds, so that by the predicate consistency principle we have $S \vdash \mathsf{false}$ also, i.e. there exists a proof of 'false' from S. Since any proof from S can involve at most finitely many formulae of S, there must exist a finite subset S' of S such that $S' \vdash \mathsf{false}$ holds, and so by the predicate consistency principle $S' \models \mathsf{false}$ must hold. That is, S' is not satisfiable, contradicting our initial hypothesis that every finite subset of S is satisfiable.

2.2.7.3 Some Other Consequences of the Gödel Completeness Theorem

Skolem Normal Form Let S be a countable (i.e. finite or denumerable) collection of syntactically well-formed predicate sentences. Putting each of these formulae into prenex normal form gives an equivalent set S' of formulae, so that if S has a model (i.e. it is consistent) so does S'. We will now describe a second normal form, called the *Skolem normal form*, into which the formulae of S' can be put. We will see that if S^{**} denotes the set of formulae in Skolem normal form derived from S', then S^{**} is consistent if and only if S' (and S) is consistent. However, the formulae of S^{**} are generally not equivalent to the formulae of S' from which they derive. Thus S^{**} and S' (and S) are only *equiconsistent*, not *equivalent*.

By definition, a formula in prenex normal form is in Skolem normal form if and only if its prefixed list of quantifiers contains no existential quantifiers. To derive the Skolem normal form of a formula F in S', which must already be in prenex normal form, suppose that F has the form

$$\big(\forall x_1, \ldots, x_k \mid (\exists y \mid G)\big). \tag{2.11}$$

Introduce a new function symbol f of k variables, along with a statement of the form

$$\big(\forall x_1, \ldots, x_k \mid G(y \hookrightarrow e)\big), \tag{2.12}$$

where $G(y \hookrightarrow e)$ is derived from G by replacing every free appearance of the variable y in G by an appearance of the subexpression $e = f(x_1, \ldots, x_k)$. Let S_1 be the result of adding (2.12) to S'. We have seen above that S_1 is a conservative extension of S'. Hence if $S' \vdash \mathsf{false}$ is false, so is $S'_1 \vdash \mathsf{false}$, and conversely. That is, S' and S_1 are equiconsistent.

Let S^* be the set of statements obtained by dropping (2.11) from S_1. We shall show that S' and S^* are equiconsistent. But in S^* the existentially quantified statement (2.11) has been replaced by (2.12) which has one fewer existential quantifier. It should be clear that by repeating this step as often as necessary, we can eliminate all existential quantifiers from our original set of statements, introducing function symbols in their stead. The resulting set of statements is the Skolem normal form of our original set. To prove that S' and S^* are equiconsistent, note first of all that, as we have already noted, S^* is consistent if S' is consistent. Suppose conversely that S^* is consistent. We can deduce $G(y \hookrightarrow e)$ from (2.12) by k successive applications of predicate axiom (v) and the rule of modus ponens. More specifically, we have

$$\big(\forall x_1, \ldots, x_k \mid G(y \hookrightarrow e)\big) \vdash G(y \hookrightarrow e).$$

But since

$$\vdash (\forall y \mid \neg G) \to \big(\neg G(y \hookrightarrow e)\big)$$

by the same axiom (v), it follows that

$$\big(\forall x_1, \ldots, x_k \mid G(y \hookrightarrow e)\big) \vdash \neg(\forall y \mid \neg G).$$

Thus by predicate axiom (iii) we have

$$\big(\forall x_1, \ldots, x_k \mid G(y \hookrightarrow e)\big) \vdash (\exists y \mid G)$$

and so, by repeated application of the rule of generalization, we obtain

$$\big(\forall x_1, \ldots, x_k \mid G(y \hookrightarrow e)\big) \vdash \big(\forall x_1, \ldots, x_k \mid (\exists y \mid G)\big).$$

The deduction theorem now implies

$$\vdash \big(\forall x_1, \ldots, x_k \mid G(y \hookrightarrow e)\big) \to \big(\forall x_1, \ldots, x_k \mid (\exists y \mid G)\big)$$

so that

$$S^* \vdash \big(\forall x_1, \ldots, x_k \mid (\exists y \mid G)\big).$$

This implies that exactly the same formulae can be derived from S_1 and S^*, so that these two sets of formulae are equiconsistent. Hence S' and S^* are equiconsistent, as required.

The Herbrand Theorem Herbrand's theorem, which gives a *semi-decision procedure* for the satisfiability of sets of predicate formulae given in Skolem normal form, can be stated as follows.

Theorem 2.4 (Herbrand) *Let S be a countable collection of predicate sentences, all having Skolem normal form. Let D be the set of all function symbols appearing in the formulae of S. Let SC be the set of individual constants (function symbols of zero variables) appearing in the formulae of S. (If there are no such constants, let SC consist of just one artificially introduced individual constant, distinct from all the other symbols in D.) Let T be the set of all terms which can be generated from the constants in SC using the function symbols appearing in formulae of S. Let S' be the set of formulae generated from S by stripping off their quantifiers and substituting terms in T for the variables of the resulting formulae in all possible ways. Then the set S is consistent if and only if every finite subset of S' is consistent when regarded as a collection of propositional formulae in which two atomic formulae correspond to the same propositional variable if and only if they are syntactically identical.*

Proof This is just the Corollary of the Gödel completeness theorem stated above, in the special case in which the formulae of S have Skolem normal form, i.e. they contain no existential quantifiers. For in this case the construction we have used to prove that Theorem and Corollary generates no new constant symbols. □

Herbrand's theorem is often used as a technique for searching automatically for predicate-calculus proofs. If none of the formulae concerned have any free variables, we can show that a predicate formula F follows from a set S of such formulae by adjoining the negative of F to S, then putting all the resulting formulae into Skolem normal form, and finally searching for the propositional contradiction of whose existence Herbrand's theorem assures us.

As a very simple example, consider the predicate theorem

$$\big(\exists y \,\big|\, \big(\forall x \mid P(x, y)\big)\big) \rightarrow \big(\forall x \,\big|\, \big(\exists y \mid P(x, y)\big)\big) \tag{2.13}$$

whose negation is

$$\big(\exists y \,\big|\, \big(\forall x \mid P(x, y)\big)\big) \,\&\, \big(\exists x \,\big|\, \big(\forall y \mid \neg P(x, y)\big)\big), \tag{2.14}$$

or, in Skolem normal form,

$$\big(\forall x \mid P(x, B)\big) \,\&\, \big(\forall y \mid \neg P(A, y)\big).$$

A substitution then gives the propositional contradiction $P(A, B)$ & $(\neg P(A, B))$, showing the impossibility of the negated statement (2.14), and so confirming the universal validity of (2.13).

A very large literature has developed concerning optimization of searches of this kind. Some of the resulting search techniques will be reviewed in Chap. 4.

2.3 Predicate Calculus with Equality as a Built-in

The simplicity of the equality relationship and its continual occurrence in mathematical arguments make it appropriate to extend the predicate calculus as defined above to a slightly larger version in which equality is a built-in. Syntactically we have only to make '=' a reserved symbol; semantically we need to introduce axioms for equality strong enough for the Gödel completeness theorem to remain valid. The following axioms suffice.

The axioms of the *equality-extended predicate calculus* are all the axioms of the (ordinary) predicate calculus (cf. Definition 2.7), plus

(vi) Any formula of the form

$$\big(\forall x, y, z \mid x = x \;\&\; \big((x = y) \to (y = x)\big) \;\&\; \big((x = y \;\&\; y = z) \to (x = z)\big)\big).$$

(vii) Any formula of the form

$$\big(\forall x, y \mid (x = y) \to \big(f(x_j \hookrightarrow x) = f(x_j \hookrightarrow y)\big)\big),$$

where f is a k-adic functional expression $f(x_1, \ldots, x_k)$, and $f(x_j \hookrightarrow x)$ (resp. $f(x_j \hookrightarrow y)$) is the result of replacing the jth variable in it by an occurrence of x (resp. y).

(viii) Any formula of the form

$$\big(\forall x, y \mid (x = y) \to \big(P(x_j \hookrightarrow x) \leftrightarrow P(x_j \hookrightarrow y)\big)\big),$$

where P is a k-adic predicate expression $P(x_1, \ldots, x_k)$, and $P(x_j \hookrightarrow x)$ (resp. $P(x_j \hookrightarrow y)$) is the result of replacing the jth variable in it by an occurrence of x (resp. y).

No new rules of inference are added.

The notion of 'model' is extended to this slightly enlarged version of the predicate calculus by agreeing that

(xi) If the formula F is of the form '$t_1 = t_2$', then

$$\mathsf{Val}(I, A, F) = \textbf{if } \mathsf{Val}(I, A, t_1) = \mathsf{Val}(I, A, t_2) \textbf{ then } 1 \textbf{ else } 0 \textbf{ end if},$$

for every interpretation framework $(\mathscr{U}, I, A)$.

That is, the predicate which models the equality sign is simply the standard predicate of equality.

As before we want to show that the added predicate axioms evaluate to 1 in every model. This is clear for (vi), since it simply states the standard properties of equality. Similarly, since replacement of the arguments of any set-theoretic mapping by an equal argument never changes the map value, (vii) and (viii) must evaluate to 1 in any model.

Additionally we can show that the Gödel completeness theorem carries over to our extended predicate calculus. For this, we argue as follows. If $(\mathscr{U}, I, A)$ is an interpretation framework covering a set S of sentences in our extended calculus, then it follows as previously that if $\mathsf{Val}(I, A, F) = 1$ for each F in S, then $\mathsf{Val}(I, A, G) = 1$ for every G such that $S \vdash G$. Hence, as previously, if such a set S has a model it is consistent. Suppose conversely that S is consistent. Add the equality axioms (vi–viii) to S (this preserves consistency since only axioms are added to S) and proceed as above to build the sets SC, SUF, SNQ, and the list LEF. Then the collection of statements in SNQ must be propositionally consistent, and so must have a propositional model V for which every statement in SNQ takes on the value 'true'. It was seen above that this gives a model $(\mathscr{U}, I, A)$ of all the statements in our collection, with universe $\mathscr{U}$ equal to the set of all terms formed from the constants in SC using the function symbols appearing in formulae of S. This is not quite a model of S in the sense required when we take '=' as a built-in predicate symbol which must be modeled by the standard equality operator, since there may well exist formulae of the form $t_1 = t_2$ such that $\mathsf{Val}(I, A, t_1 = t_2) = 1$ even though t_1 and t_2 are syntactically distinct. However, the binary relationship

$$R(t_1, t_2) = \big(\mathsf{Val}(I, A, t_1 = t_2) = 1\big) \tag{2.15}$$

between terms of $\mathscr{U}$ must be an equivalence relation, since whenever terms t_1, t_2 and t_3 are generated we will have added all the assertions

$$t_1 = t_1 \;\&\; \big((t_1 = t_2) \to (t_2 = t_1)\big) \;\&\; \big((t_1 = t_2 \;\&\; t_2 = t_3) \to (t_1 = t_3)\big)$$

to our collection. Moreover, since in the same situation statements like

$$\big(t_1 = t_2 \to \big(f(\cdots t_1 \cdots) = f(\cdots t_2 \cdots)\big)\big)$$

and

$$\big(t_1 = t_2 \to \big(P(\cdots t_1 \cdots) \leftrightarrow P(\cdots t_2 \cdots)\big)\big)$$

will have been added to our collection for all function and predicate symbols, the terms must always be equivalent whenever their lead function symbols are the same and their arguments are equivalent, and also we must have $\mathsf{Val}(I, A, P(\cdots t_1 \cdots)) = \mathsf{Val}(I, A, P P(\cdots t_2 \cdots))$ for atomic formulae when their lead function symbols are the same and their arguments are equivalent. Therefore we can form a model of our set of statements by replacing the universe $\mathscr{U}$ by the set $\mathscr{U}'$ of equivalence classes on it defined by the equivalence relation (2.15), and in this new model the symbol '=' is represented by the standard equality operation. This concludes our proof that the Gödel completeness theorem carries over to our extended predicate calculus.

2.4 Set Theory as an Axiomatic Extension of Predicate Calculus

In most of the present book we take a rather free version of set theory (perhaps this should be called 'brutal' set theory) as basic, and use it to hurry onward to our main goal of proving the long list of theorems found in Chap. 5. The standard treatment of set theory ties it more carefully to predicate calculus. Specifically, to ensure applicability of the foundational results presented earlier in this chapter, set theory is cast as a collection of predicate axioms. In this form it is customarily referred to as Zermelo–Fraenkel set theory (ZF) if no version of the axiom of choice is necessarily included, or ZFC if an axiom of choice is present. Here is the standard list of ZFC axioms.

2.4.1 Zermelo–Fraenkel Theory with the Axiom of Choice

(1) **(Axiom of extension)** $(\forall s,\ t \mid (s = t) \leftrightarrow (\forall x \mid (x \in s) \leftrightarrow (x \in t)))$.

(2) **(Axioms of elementary sets)** There is an empty set $\emptyset$; for each set t there is a set $\mathsf{Singleton}(t)$ whose only member is t; if s and t are sets then there is a set $\mathsf{Unordered\_pair}(s, t)$ whose only members are s and t. That is, we have

$$\big(\forall s \mid \neg(s \in \emptyset)\big),$$
$$\big(\forall t,\ u \mid \big(u \in \mathsf{Singleton}(t)\big) \leftrightarrow (u = t)\big),$$
$$\big(\forall s,\ t,\ u \,\big|\, \big(u \in \mathsf{Unordered\_pair}(s, t)\big) \leftrightarrow \big((u = s) \vee (u = t)\big)\big).$$

(3) **(Axiom of power set)** To every set A there corresponds a set $\mathscr{P}(A)$ whose members are precisely the subsets of A:

$$\big(\forall s,\ t \,\big|\, \big(s \in \mathscr{P}(t)\big) \leftrightarrow \big(\forall x \mid (x \in s) \leftrightarrow \big(\forall y \mid (y \in x) \rightarrow (y \in t)\big)\big)\big).$$

(4) **(Axiom of union)** To every set A there corresponds a set $\bigcup A$ whose members are precisely those elements belonging to elements of A:

$$\Big(\forall s,\ t \,\Big|\, \Big(s \in \bigcup t\Big) \leftrightarrow \big(\exists x \mid (x \in t)\ \&\ (s \in x)\big)\Big).$$

(5) **(Axiom of infinity)** There is at least one set Inf such that

$$(\emptyset \in \mathsf{Inf})\ \&\ \big(\forall s \mid (s \in \mathsf{Inf}) \rightarrow \big(\mathsf{Singleton}(s) \in \mathsf{Inf}\big)\big).$$

(6) **(Axiom of regularity)**

$$\neg\big(\exists x \mid (x \neq \emptyset)\ \&\ \big(\forall y \mid (y \in x) \rightarrow \big(\exists z \mid (z \in x)\ \&\ (z \in y)\big)\big)\big).$$

(7) **(Axiom schema of subsets)** If $F(y, z_1, \ldots, z_n)$ is any syntactically valid formula of the language of ZF that has no free variables other than those shown, and neither x nor z occur in the list $y, z_1, \ldots, z_n$, then

$$\big(\exists z \mid \big(\forall y \mid (y \in z) \leftrightarrow \big((y \in x)\ \&\ F(y, z_1, \ldots, z_n)\big)\big)\big)$$

is an axiom. Here and below, a formula is said to be a formula of the language of ZF if it is formed using only the built-in symbols of predicate calculus (i.e. the propositional operators, $\forall, \exists, =$) plus the membership operator. (Note that in stating this axiom, we mean to assert the formula which results by quantifying it universally over all the free variables $z_1, \ldots, z_n$.)

(8) **(Axiom schema of replacement)** If $F(u, v, z_1, \ldots, z_n)$ is any syntactically valid formula of the language of ZF that has no free variables other than those shown, and neither u nor v occur in the list $z_1, \ldots, z_n$, then

$$\big(\forall u, v_1, v_2 \,\big|\, \big(\big(F(u, v_1, z_1, \ldots, z_n) \,\&\, F(u, v_2, z_1, \ldots, z_n)\big) \to (v_1 = v_2)\big)\big)$$
$$\to \big(\forall b \,\big|\, \big(\exists c \,\big|\, \big(\forall y \mid (y \in c) \leftrightarrow \big(\exists x \mid (x \in b) \,\&\, F(x, y, z_1, \ldots, z_n)\big)\big)\big)\big)$$

is an axiom. (Here again, in stating this axiom, we mean to assert the formula which results by quantifying it universally over all the free variables $z_1, \ldots, z_n$.)

This statement is obscure enough for a brief clarifying discussion of its equivalent in our informal version of set theory to be helpful. In that less formal system we would proceed by defining an auxiliary 'Skolem' function h satisfying

$$\big(\forall x, z_1, \ldots, z_n \,\big|\, \big(\exists y \mid F(x, y, z_1, \ldots, z_n)\big)$$
$$\leftrightarrow F\big(x, h(x, z_1, \ldots, z_n), z_1, \ldots, z_n\big)\big).$$

Then, since the replacement axiom assumes that $F(x, y, z_1, \ldots, z_n)$ defines y uniquely in terms of x and $z_1, \ldots, z_n$, we have

$$\big(\forall x, y, z_1, \ldots, z_n \mid F(x, y, z_1, \ldots, z_n) \to \big(y = h(x, z_1, \ldots, z_n)\big)\big),$$

and so the set c whose existence is asserted by the axiom of replacement can be written in our 'working' version of set theory as

$$\big\{h(x, z_1, \ldots, z_n) : x \in b \mid F\big(x, h(x, z_1, \ldots, z_n), z_1, \ldots, z_n\big)\big\}.$$

This 'set-former' expression is the form in which such constructs will almost always be written.

(9) **(Axiom of choice)**

$$\big(\forall x \,\big|\, \big(\exists f \mid \mathsf{Svm}(f) \,\&\, \big(\mathsf{domain}(f) = x\big)$$
$$\&\, \big(\forall y \,\big|\, \big((y \in x) \,\&\, (y \neq \emptyset)\big) \to \big(f[y] \in y\big)\big)\big)\big).$$

Note that this form of the axiom of choice is weaker than the assumption concerning 'arb' which our 'brutal' set theory uses in its place. Specifically, while 'arb' is a universal choice function applicable to any non-null set, the axiom of choice just stated provides a separate such choice function for each set of sets.

Most axioms appear in Skolemized version in the above list. Other authors prefer to write those in unskolemized form, e.g. to write our axiom $(\forall s \mid \neg(s \in \emptyset))$ in the form

$$\big(\exists z \,\big|\, \big(\forall s \mid \neg(s \in z)\big)\big).$$

Similarly the axiom of union will often be written as

$$\big(\forall t \,\big|\, \big(\exists u \,\big|\, \big(\forall s \mid (s \in u) \leftrightarrow \big(\exists x \mid (x \in t)\ \&\ (s \in x)\big)\big)\big)\big).$$

The main respects in which the ZFC formulation of set theory differs from our 'brutal' version is that no built-in set-former construct is provided, nor are 'transfinite recursive' definitions like those freely allowed in our version of set theory. An issue of relative consistency therefore arises: can our version of set theory be reduced to ZFC in some standard way, or, if ZFC is assumed to be consistent, can it be demonstrated that our 'brutal' version is consistent also?

2.4.2 Concerning the Consistency of ZFC and Various Interesting Extensions of It

To open a discussion of this problem we first consider the general question of consistency for set-theoretic axioms like the ZFC axioms. Since equality can be treated as an operator of logic, these axioms involve only one non-logical symbol, the predicate symbol '$\in$'. The Gödel completeness theorem tells us that the ZFC axioms are consistent if and only if they have a model. How can such models be found? Are there many of them having an interesting variety of properties, or just a few? Since von Neumann's 1928 paper on the axioms of set theory and Gödel's 1938 work on the continuum hypothesis, many profound studies have addressed these questions. We can get some initial idea of the issues involved by looking a bit more closely at the hereditarily finite sets. We will see that these are of interest in the present context since they model all the axioms of set theory other than the axiom of infinity.

2.4.2.1 Basic Facts Concerning Hereditarily Finite Sets

In intuitive terms, the '*hereditarily finite*' sets s are those which can be constructed by using the pair formation operation $\{x, y\}$ and union operation $x \cup y$ repeatedly, starting from the null set $\{\}$ (same as $\emptyset$). Any such set has a string representation r consisting of a properly matched arrangement of opening brackets '{' and closing brackets '}', 'properly matched' in the sense that there are equally many opening and closing brackets, and that no initial substring of r contains more closing than opening brackets. Moreover, the string representation r of any such set is indecomposable, in the sense that no initial substring of r is properly matched. Three examples are

$$\{\} \qquad \big\{\{\}\big\} \qquad \big\{\{\}\{\{\}\}\big\}.$$

The 'height' of any such set is one less than the maximum depth of bracket nesting in its string representation. For example, the three sets just displayed have heights 0, 1, and 2, respectively. The general transfinite induction techniques described in the preceding section make it possible to prove that the hereditarily finite sets are precisely those sets which are finite and all of whose elements are themselves hereditarily finite; this point is discussed in greater detail in Sect. 4.3.10 and in Chap. 6.

Hereditarily finite sets can be represented in many ways by computer data structures which allow the basic operations on them, namely $\{x, y\}$, $x \cup y$, and $x \in y$, to be realized by simple code fragments, and therefore allow translation of set-former expressions and recursive function definitions of all kinds into computer programs. One way of doing this is to make direct use of string representations like those just displayed. To this end, note that each properly matched arrangement of brackets is a concatenation of one or more indecomposable properly matched arrangements of brackets, and that every indecomposable arrangement has the form $\{s\}$ where s itself is properly matched. Moreover the decomposition of any properly matched arrangement of brackets into indecomposable properly matched substrings is unique. (The reader is invited to prove these elementary facts, and to describe an algorithm for separating any properly matched arrangement of brackets into its indecomposable parts.)

It follows from the facts just stated that each hereditarily finite set t has a string representation, itself indecomposable, of the form

$$\{s_1\, s_2\, \cdots\, s_m\}, \tag{2.16}$$

where each of the s_j is properly matched and indecomposable, and where all these s_j, which are simply the string representations of the elements of t, are distinct. We can make this string representation unique by insisting that the s_j be arranged in order of increasing length, members having string representations of the same length then being arranged in alphabetical order of their representations. We can call a string representation (2.16) having these properties at every recursive level (and in which all the s_j are distinct at every level) a 'nicely arranged' properly matched arrangement of brackets.Then every hereditarily finite set has a unique string representation of this kind, and conversely every nicely arranged properly matched arrangement of brackets represents a unique set. Hence these arrangements give an explicit, 1-1 representation of the family of all hereditarily finite sets.

In this representation, the two elementary operations $\{x, y\}$ and $x \cup y$ which suffice for construction of all such sets have the following simple implementations. The representation of $\{x, y\}$ is obtained by taking the representations s_x and s_y of x and y, respectively, checking them for equality and eliminating one of them if they are equal, arranging them in order of length (or alphabetically if their lengths are equal), and forming the string $\{s_x\, s_y\}$ (or simply $\{s_x\}$ if s_x and s_y are identical). To compute the standard string representation of $x \cup y$, let $\{\, s_1\, s_2\, \cdots\, s_m\,\}$ and $\{\, t_1\, t_2\, \cdots\, t_n\,\}$ be the standard string representations of x and y, respectively. Then form the concatenation

$$s_1 s_2 \cdots s_m t_1 t_2 \cdots t_n,$$

rearrange its indecomposable parts in the standard order described above, eliminate duplicates, and enclose the result in an outermost final pair of brackets.

In this, or any other convenient representation, it is easy to construct a code fragment which will calculate the value of any set former of the type we allow, for example

$$\{e(x) : x \in s \mid P(x)\},$$

provided that s is hereditarily finite, and that e is any set-valued expression and $P(x)$ any predicate expression which can be calculated by procedures which have already been constructed. For this, we have only to set up an iterative loop over all the elements of s, and use an operation which calculates $e(x)$ for each element x of s satisfying $P(x)$ and then inserts all such elements into an initially empty set, eliminating duplicates.

The powerset operation $\mathscr{P}(s)$ (set of all subsets of s) satisfies the recursive relationship

$$\begin{aligned}\mathscr{P}(s) = {} & \textbf{if } s = \emptyset \textbf{ then } \{\emptyset\} \\ & \textbf{else } \mathscr{P}(s \setminus \{\mathsf{arb}(s)\}) \cup \{x \cup \{\mathsf{arb}(s)\} : x \in \mathscr{P}(s \setminus \{\mathsf{arb}(s)\})\} \\ & \textbf{end if}\end{aligned}$$

which can be used to calculate $\mathscr{P}(s)$ recursively for each hereditarily finite s. This makes it possible to calculate set formers of the second allowed form

$$\{e(x) : x \subseteq s \mid P(x)\},$$

by translating them into

$$\{e(x) : x \in \mathscr{P}(s) \mid P(x)\}.$$

Set formers involving multiple bound variables, for example

$$\{e(x, y, z) : x \in s,\ y \in a(x),\ z \in b(x, y) \mid P(x, y, z)\},$$

can be calculated in much the same way using multiply nested loops, provided that all the sets which appear are hereditarily finite and that e, a, and b are set-valued expressions, and $P(x, y, z)$ a predicate expression, which can be calculated by procedures which have already been constructed. Similar loops can be used to calculate existentially and universally quantified expressions like

$$(\forall x \in s,\ y \in a(x),\ z \in b(x, y) \mid P(x, y, z))$$

and

$$(\exists x \in s,\ y \in a(x),\ z \in b(x, y) \mid P(x, y, z)),$$

or such simpler quantifiers as

$$(\forall x \in s \mid P(x)) \quad \text{and} \quad (\exists x \in s \mid P(x)).$$

Note, however, that the predicate calculus in which we work also allows quantifiers involving bound variables not subject to any explicit limitation, for example

$$\big(\forall x \mid P(x)\big) \quad \text{and} \quad \big(\exists x \mid P(x)\big).$$

Since translation of expressions of this form into a programmed loop would require iteration over the infinite collection of all hereditarily finite sets, we can no longer claim that the values of these unrestricted iterators are effectively calculable. Thus they represent a first step into the more abstract world of the actually infinite, where symbolic reasoning must replace explicit calculation.

All the kinds of definition we allow translate just as readily into computer codes as long as only hereditarily finite sets are considered. Algebraic definitions like

$$\bigcup x =_{\text{Def}} \{z : y \in x \,\&\, z \in y\}$$

translate directly into procedures whose body consists of a single nested iteration. Recursive definitions like

$$\begin{aligned}\mathsf{enum}(X, S) =_{\text{Def}} \textbf{if} \quad & S \subseteq \big\{\mathsf{enum}(y, S) : y \in X\big\} \quad \textbf{then} \quad S \\ & \textbf{else } \mathsf{arb}\big(S \setminus \big\{\mathsf{enum}(y, S) : y \in X\big\}\big) \ \textbf{end if}\end{aligned}$$

translate just as directly into recursive procedures. Thus, as long as we confine ourselves to hereditarily finite sets, the whole of the set theory in which we work (excepting only unrestricted quantifiers of the kind shown above) can be thought of both as a language for the description of mathematical relationships and as an implementable (indeed, implemented) programming language for actual manipulation of a convenient class of finite objects. This parallelism between language of deduction and language of computation will be explored more deeply in Chaps. 4 and 6.

We can summarize the preceding discussion in the following way. All hereditarily finite sets can be given explicit finite representations, so that these sets constitute a 'universe of computation' in which all of the properties we assume for sets can be checked by explicit computation, at least in individual cases. We will see below that the collection of hereditarily finite sets models all the axioms of set theory, save one: there is no infinite set, for example no hereditarily finite set t having the property

$$t \neq \emptyset \quad \& \quad \big(\forall x \in t \mid \{x\} \in t\big)$$

which we will use as our axiom of infinity. By including this statement in our collection of axioms we cross from the world of computation defined by the hereditarily finite sets into a more abstract world of objects which can no longer be enumerated explicitly but which are known only through the statements about them that we can deduce formally, i.e. as elements of a world of formal computation, whose main elementary property is simply its formal consistency. Nevertheless, mathematical experience has shown that the statements that we can prove about the objects of this abstract world are both beautiful and extremely useful tools for deriving many properties of hereditarily finite sets which it would be harder or impossible to prove if we refused to enlarge our universe of discourse to allow free reference to infinite sets.

2.4.2.2 Hereditarily Finite Sets: Formal Definition Within General Set Theory

Hereditarily finite sets can be defined formally in either of two ways: either as all sets satisfying a predicate Is_HF, or as all the members of a set HF. The predicate Is_HF is defined in the following recursive way (we continue to designate the set of all integers by $\mathbb{N}$):

$$\mathsf{Is\_HF}(x) \leftrightarrow_{\mathrm{Def}} \big((\#x \in \mathbb{N}) \,\&\, \big(\forall y \in x \mid \mathsf{Is\_HF}(y)\big)\big).$$

To define the corresponding set HF (thereby showing that the collection of all x satisfying $\mathsf{Is\_HF}(x)$ is really a set), a bit more work is needed. We proceed as follows. Begin with the following recursive definition (informally speaking, this defines the collection of all sets of 'rank x'):

$$\mathsf{HF\_}(x) =_{\mathrm{Def}} \textbf{if } x = \emptyset \textbf{ then } \emptyset \textbf{ else } \bigcup \big\{ \mathscr{P}\big(\mathsf{HF\_}(y)\big) : y \in x \big\} \textbf{ end if }.$$

It is easily proved by induction that

$$\big(\forall y \in \mathsf{HF\_}(x) \mid \mathsf{HF\_}(x) \supseteq y\big).$$

Indeed, if there exists an x for which '$\mathsf{HF\_}(x) \supseteq z$' is false for some z in $\mathsf{HF\_}(x)$, there exists a smallest such x, which, after renaming, we can take to be x itself. Then there is a u such that $z \in \mathsf{HF\_}(x)$, $u \in z$, $u \notin \mathsf{HF\_}(x)$. Since $z \in \mathsf{HF\_}(x)$, we have

$$z \in \bigcup \big\{ \mathscr{P}\big(\mathsf{HF\_}(y)\big) : y \in x \big\},$$

so $z \in \mathscr{P}(\mathsf{HF\_}(y))$ for some $y \in x$, i.e. $z \subseteq \mathsf{HF\_}(y)$ for some $y \in x$. Then $u \in \mathsf{HF\_}(y)$ for some $y \in x$. Since x has no member y for which

$$\big(\forall w \in \mathsf{HF\_}(y) \mid \mathsf{HF\_}(y) \supseteq w\big)$$

is false, it follows that $\mathsf{HF\_}(y) \supseteq u$, so $u \in \mathscr{P}(\mathsf{HF\_}(y))$, and therefore

$$u \in \bigcup \big\{ \mathscr{P}\big(\mathsf{HF\_}(y)\big) : y \in x \big\},$$

i.e. $u \in \mathsf{HF\_}(x)$, proving our claim. Note also that the function HF_ is increasing in its parameter, in the sense that if $y \in x$, then $\mathsf{HF\_}(x) \supseteq \mathsf{HF\_}(y)$. Indeed if u is an element of $\mathsf{HF\_}(y)$, then $\{u\} \in \mathscr{P}(\mathsf{HF\_}(y))$, so

$$\{u\} \in \bigcup \big\{ \mathscr{P}\big(\mathsf{HF\_}(y)\big) : y \in x \big\},$$

and therefore $\{u\} \in \mathsf{HF\_}(x)$, so by what we have just proved $u \in \mathsf{HF\_}(x)$.

In what follows we also need the fact that

$$\big(\forall n \in \mathbb{N} \mid \#\mathsf{HF\_}(n) \in \mathbb{N}\big),$$

i.e. that all the sets $\mathsf{HF\_}(n)$ are themselves finite. To prove this, suppose that it fails for some smallest n. Then

$$\mathsf{HF\_}(n) = \bigcup \big\{ \mathscr{P}\big(\mathsf{HF\_}(m)\big) : m \in n \big\},$$

all the sets $\mathsf{HF\_}(m)$ for which $m \in n$ are finite, and so are their power sets. Thus $\mathsf{HF\_}(n)$ is the union of a sequence of sets, each of finite cardinality, over a domain of cardinality less than $\mathbb{N}$ (i.e. of finite cardinality). Hence $\mathsf{HF\_}(n)$ is itself finite, i.e. $\#\mathsf{HF\_}(n)$ belongs to $\mathbb{N}$, as asserted.

Now we can define the set HF by

$$\mathsf{HF} =_{\text{Def}} \bigcup \big\{ \mathsf{HF\_}(n) : n \in \mathbb{N} \big\}. \tag{2.17}$$

To come to the desired goal we must prove that

$$\big(\forall y \mid \mathsf{Is\_HF}(y) \leftrightarrow y \in \mathsf{HF}\big).$$

This can be done as follows. Suppose that $y \in \mathsf{HF}$. Then we have $y \in \mathsf{HF\_}(n)$ for some $n \in \mathbb{N}$. To prove that $\mathsf{Is\_HF}(y)$, suppose that this is false, and, proceeding inductively, that n is the smallest element of $\mathbb{N}$ for which $\mathsf{HF\_}(n)$ has an element y such that $\mathsf{Is\_HF}(y)$ is false. Then, since

$$y \in \bigcup \big\{ \mathscr{P}\big(\mathsf{HF\_}(m)\big) : m \in n \big\},$$

we have $y \in \mathscr{P}(\mathsf{HF\_}(m))$ for some $m \in n$. All the elements u of y are therefore elements of $\mathsf{HF\_}(m)$, and so satisfy $\mathsf{Is\_HF}(u)$. We have also proved that $\mathsf{HF\_}(m)$ is finite, so all its subsets are finite, and therefore $\#y \in \mathbb{N}$, proving that $\mathsf{Is\_HF}(y)$, a contradiction implying that

$$(y \in \mathsf{HF}) \rightarrow \mathsf{Is\_HF}(y)$$

for all y.

Suppose conversely that $\mathsf{Is\_HF}(x)$, and that $x \notin \mathsf{HF}$. Proceeding inductively, we can suppose that x is a minimal element with these properties, i.e. that $y \in \mathsf{HF}$ for each $y \in x$. Then it follows from (2.17) that for each y in x there is an $n = n(y)$ in $\mathbb{N}$ for which $y \in \mathsf{HF\_}(n(y))$. But then since x is finite by definition of $\mathsf{Is\_HF}(x)$, the maximum m of all these $n(y)$ is finite, so every y in x belongs to $\mathsf{HF\_}(m)$ since the sets $\mathsf{HF\_}(m)$ clearly increase with their parameter m. Therefore $x \in \mathscr{P}(\mathsf{HF\_}(m))$, $x \in \mathsf{HF\_}(m+1)$, and $x \in \mathsf{HF}$, a contradiction implying that

$$\mathsf{Is\_HF}(y) \rightarrow (y \in \mathsf{HF})$$

for all y, which leads to the desired conclusion.

It is easily seen that HF is a model of all the ZFC axioms *other than the axiom of infinity*. To show this, we simply need to check that all these axioms remain valid if we interpret all quantifiers as extending over the set HF rather than over the 'universe of all sets' that the initial ZFC axioms assume. This can be done as follows.

(1) The axiom of extension remains true since HF is *transitive*, i.e. every member of a member of HF belongs to HF. (2) The null set, singleton, and unordered pair constructions take elements of HF into themselves since they construct finite sets all of whose elements are drawn from HF. (3) The power set axiom remains valid since every subset of an hereditarily finite set is hereditarily finite, and for s in HF, $\mathscr{P}(s)$ consists only of such elements and also is finite. (4) The union set axiom remains valid since every member of a member of $\bigcup s$, where s is an hereditarily finite set, is hereditarily finite, and for $s \in \mathsf{HF}$, $\bigcup s$ is the union of finitely many sets and so is finite. (5) The axiom of infinity fails. (6) The axiom of regularity clearly remains true, since each $z \in \mathsf{HF}$ has the same members as an element of HF that it does as a set. (7) The axiom schema of subsets, which in informal terms asserts the existence of the set $y = \{u : u \in x \mid F(x, z_1, \ldots, z_n)\}$ for every x and $z_1, \ldots, z_n$, remains true since the y whose existence it asserts is a subset of the x which it assumes, and so must be hereditarily finite if x is hereditarily finite. (8) In informal terms, the axiom schema of replacement asserts the existence of the set $y = \{u : x \in b \mid F(x, u, z_1, \ldots, z_n)\}$ for every b and $z_1, \ldots, z_n$ if the predicate F defines u uniquely in terms of x and $z_1, \ldots, z_n$. This remains true if only hereditarily finite sets are allowed, since if b is finite and each u is required to be hereditarily finite the set of whose existence it asserts is a finite set of elements, each of which is hereditarily finite, and so must be hereditarily finite. (9) The axiom of choice remains true since the f whose existence it asserts is a single-valued map whose pairs have their first components in x and their second components in $\bigcup x$: assuming that $x \in \mathsf{HF}$, each such pair plainly belongs to HF and therefore, since f consists of finitely many such pairs, we conclude that $f \in \mathsf{HF}$. (If $\emptyset \in x$, we can carry out a similar argument, after replacing the image $f(\emptyset)$ by $\emptyset$.)

2.4.2.3 Large Cardinal Axioms

The preceding observations concerning the set HF suggest that it may be possible to find a model of set theory, which would imply the consistency of set theory, by replacing $\mathbb{N}$, the smallest infinite cardinal, by something larger in the crucial formula (2.17) seen above. If this is done, the argument that we have given can be shown to go through almost without change for any cardinal having the two properties of $\mathbb{N}$ used in the argument. The following definition gives names to these properties:

Definition 2.9 A non-null cardinal number N is *inaccessible* if (a) Any set of cardinals, all less than N, which has a cardinality smaller than N also has a supremum less than N. (Cardinals having this property are called *regular* cardinals.) (b) If M is a cardinal less than N, then 2^M (which is $\#\mathscr{P}(M)$ by definition) is less than N. (Cardinals which have this property are called *strong limit* cardinals.)

Note that the set $\mathbb{N}$ of integers is inaccessible according to this definition. Intuitively speaking, a cardinal number N is inaccessible if it cannot be constructed from smaller cardinals using any 'explicit' set-theoretic operation, so that the very

existence of N would seem to involve some new assumption, in the same way that assuming the existence of an infinite set takes a step beyond anything that follows from the properties of hereditarily finite sets $x \in \mathsf{HF}$.

If we make the following quite straightforward definition, which simply generalizes the preceding construction of HF to arbitrary cardinal numbers N,

Definition 2.10 $\mathscr{H}(N) =_{\text{Def}} \bigcup \{\mathsf{HF\_}(n) : n \in N\}$ for every cardinal number N,

then the preceding discussion shows that

Theorem 2.5 *If N is an inaccessible cardinal larger than $\mathbb{N}$, then $\mathscr{H}(N)$ is a model of the ZFC axioms of set theory.*

Corollary 2.2 *It there exists any inaccessible cardinal larger than $\mathbb{N}$, then the ZFC axioms have a model, and so are consistent.*

A theorem of Gödel to be proved in Chap. 6 shows that no system having at least the expressive power and proof capability of HF can be used to prove its own consistency. Thus the corollary just stated implies the following additional result:

Corollary 2.3 *Adding the assumption that there exists an inaccessible cardinal larger than $\mathbb{N}$ to the ZFC axioms allows us to construct a model of the ZFC axioms and hence implies that these axioms are consistent. Therefore the ZFC axioms cannot suffice to prove that there exists an inaccessible cardinal larger than $\mathbb{N}$.*

The situation described by this last corollary is much like that seen in the case of HF. The ZFC axioms, which include the axioms of infinity, allow us to define the infinite cardinal number $\mathbb{N}$ and so the model HF of the theory of hereditarily finite sets. The theory of hereditarily finite sets can be formalized by dropping the axiom of infinity (keeping the other axioms of ZFC, and adding a suitable principle of induction); but the resulting set of 'HF axioms' do not suffice to prove the existence of even one infinite set.

The technique for forming models of set theory seen in the preceding discussion, namely identification of some transitive set $\mathscr{H}$ in which the ZFC axioms remain true if we redefine all quantifiers to extend over the set $\mathscr{H}$ only, does not change the definition of ordinal numbers, since an element t of s is an ordinal (in the overall ZFC theory) iff its members are totally ordered by membership and each member of a member of t is a member of t. Since the collection of members of t remains the same in $\mathscr{H}$, this definition is plainly invariant. Thus the ordinal numbers of the model $\mathscr{H}$, seen from the vantage point of the overall ZFC universe, are just those ordinals which are members of $\mathscr{H}$. But the situation is different for cardinal numbers, which are defined as those ordinals O which cannot be mapped to smaller ordinals by a 1-1 mapping, i.e. those which do not satisfy

$$\mathsf{not\_cardinal}(O) \leftrightarrow_{\text{Def}} \big(\exists f \mid \mathsf{1\_1}(f)\ \&\ \mathsf{domain}(f) = O\ \&\ \mathsf{range}(f) \in O\big).$$

When we cut the whole ZFC universe of sets down to the set $\mathscr{H}$, the collection of ordinals will grow smaller, but so will the set of 1-1 mappings ('1_1s') f appearing in the formula seen above, making it unclear how the collection of cardinals (relative to $\mathscr{H}$), or the structure of this set, will change. The power set operation can also change, since for $s \in \mathscr{H}$ the power set relative to $\mathscr{H}$ is the set $\mathscr{P}(s) \cap \mathscr{H}$ of the ZFC universe. Thus properties and statements involving the power set can change meaning also. But the union set $\bigcup s$ retains its meaning. (Note also that if f is a member of $\mathscr{H}$, then the property $\mathsf{1\_1}(f)$ holds relative to $\mathscr{H}$ if and only if it holds in the ZFC universe, since it is defined by a formula quantified over the members of f, and these are the same in both contexts.)

However, in the particularly simple case in which we restrict our universe of sets to $\mathscr{H}(N)$ where N is an inaccessible cardinal, the property 'not_cardinal' does not change. This is because any 1_1 in the ZFC universe for which $\mathsf{domain}(f) \in \mathscr{H}(N)$ & $\mathsf{range}(f) \in \mathscr{H}(N)$ must itself belong to $\mathscr{H}(N)$, since it is a set of ordered pairs of elements all belonging to $\mathscr{H}(N)$, whose cardinality is at most that of $\mathsf{domain}(f)$, and so is less than N. It readily follows that the cardinals of $\mathscr{H}(N)$ are simply those cardinals of the ZFC universe which lie below N; likewise for the regular, strong limit, and inaccessible cardinals.

It follows that ZFC, plus the assumption that there are two inaccessible cardinals, allows us to construct a set $\mathscr{H}(N)$ in which there is one inaccessible cardinal (namely we take N to be the second inaccessible cardinal), and so implies the consistency of ZFC plus the axiom that there is at least one inaccessible cardinal. Generally speaking, axioms which imply the existence of many and large inaccessible cardinals imply the consistency of ZFC as extended by statements only implying the existence of fewer and smaller inaccessible cardinals, but not conversely. Thus the addition of stronger and stronger axioms concerning the existence of large cardinal numbers exemplifies a basic consequence of the incompleteness theorems presented in Chap. 6, namely that no fixed set of axioms can exhaust all of mathematics, so that significant extension of consistent systems by the addition of new axioms will always remain possible. The fact that large cardinal axioms can be formulated independently of any detailed reference to the syntax of the language of set theory makes them interesting in this regard, and so has encouraged the study of axioms which imply the existence of more and more, larger and larger, cardinal numbers.

It is worth reviewing a few of the key definitions that have appeared in such studies:

Definition 2.11 Let S be a set of cardinal numbers all of whose members are less than a fixed cardinal number N.

(i) S is said to be *closed relative to* N if the union of every sequence of elements of S whose length is less than N is a member of S.
(ii) S is said to be *unbounded in* N if every cardinal less than N is also less than some member of S.
(iii) S is said to be *thin in* N if there exists a closed unbounded set relative to N which does not intersect S.

Definition 2.12 A nonempty set F of nonempty subsets of a set S is called a *filter* on S if the intersection of any two elements of F is an element of F and any superset, included in S, of an element of F is an element of S. A filter F is an *ultrafilter* if whenever the union of finitely many subsets of S belongs to F, one of these subsets belongs to F. Given a cardinal number N, a filter F is said to be *N-complete* if whenever the union of fewer than N subsets of S belongs to F, one of these subsets belongs to F. An ultrafilter F is said to be *nontrivial* if it is not the collection of all sets having a given point p as member.

Note that if F is an N-complete filter on S, the intersection IT of any collection T of sets in F such that $\#T$ is less than N belongs to F. Indeed, S belongs to F, and if G belongs to F then $S \setminus G$ is not in F, since otherwise F would contain the null set $G \cap (S \setminus G)$. But now S is the union of IT and the collection of all complements $S \setminus G$ for $G \in T$, and since $\#T$ is less than N and F is N-complete, the union of all these complements must lie outside F, so IT must belong to F.

The following definition lists two of the various kinds of large cardinal numbers that have been considered in the literature.

Definition 2.13

(i) A cardinal number N is a *Mahlo* cardinal if it is inaccessible and the set of regular cardinals less than N is not thin.
(ii) A cardinal number N is *measurable* if there is a nontrivial N-complete ultrafilter for N.

Note that if there is a Mahlo cardinal N, then the number of inaccessible cardinals below N must be at least N. For if there were fewer, then since N is inaccessible the supremum M of all these cardinals would also be less than N. But then the set SLC of all strong limit cardinals between M and N is unbounded and closed, contradicting the assumption that N is Mahlo. Indeed, for each K between M and N, the supremum of the sequence 2^K, 2^{2^K}, ... must be a strong limit cardinal, showing that SLC is unbounded in N. Also the supremum L of any collection of strong limit cardinals must itself be a strong limit cardinal, since any L_1 less than L must plainly be less than some cardinal of the form 2^K. This shows that SLC is closed. Now, no member K of SLC can be regular, since if it were it would be inaccessible, contradicting the fact that M is the largest inaccessible below N. This shows that the set of regular cardinals below N is thin, contradicting the assumption that N is Mahlo, and so completes our proof of the fact that every Mahlo cardinal N must be the Nth inaccessible.

It follows that the assumption that there is a Mahlo cardinal is much stronger than the assumption that there is an inaccessible cardinal, since it implies that there are inaccessibly many inaccessible cardinals.

Suppose next that the cardinal number N is measurable, and let F be an N-complete nontrivial ultrafilter on N. Then any set consisting of just one point p must lie outside F (or else F would be the trivial ultrafilter consisting of all sets having p as member). Since F is N-complete, it follows that every subset of N having

fewer than N points lies outside F, and therefore so does every union of fewer than N such sets. Hence every measurable cardinal is regular. We will now show that if K is a cardinal less than N, then 2^K is less than N also, showing that every measurable cardinal is inaccessible. Suppose the contrary, so that there exists a collection CF of binary-valued functions $f(j)$ defined for all j in K, but having cardinality N, and so standing in 1-1 correspondence with N. This correspondence maps f to an N-complete nontrivial ultrafilter F' on CF. For each j in K, let $a(j)$ be that one of the two Boolean values $\{0, 1\}$ for which the set of functions $\{f \in S \mid f(j) = a(j)\}$ belongs to F'. Then, since F' is N-complete, it follows, as was shown above, that the intersection of all the sets $\{f \in S \mid f(j) = a(j)\}$ must belong to F', and so F' contains a singleton and must therefore be trivial, contrary to assumption.

This proves that any measurable cardinal N is inaccessible. Thomas Jech (whose [Jec97] is a general reference for this area of set theory) proves the much stronger result (Lemma 28.7 and Corollary, p. 313) that N must be Mahlo, and in fact must be the Nth Mahlo cardinal. He goes on to define yet a third class of cardinals, the *supercompact* cardinals (p. 408), and to show that each supercompact cardinal N must be measurable, and in fact must be the Nth measurable cardinal (Lemma 33.10 and Corollary, p. 410).

In light of the preceding, we can say that various axioms implying the existence of very many large inaccessible cardinals have been considered in the literature, with some hope that they can be used to define consistent extensions of the axioms of set theory.

The preceding discussion suggests the following transfinite recursive definition, which generalizes some of the properties of very large cardinals considered above:

$$\begin{aligned} P_x(N) \leftrightarrow_{\text{Def}} \textbf{if } x = \emptyset \textbf{ then } &\mathsf{Is\_inaccessible}(N) \\ \textbf{else } \big(\forall y \in x \mid \#\big\{M : M \in N \mid P_y(M)\big\} = N\big) &\textbf{ end if}\,. \end{aligned} \tag{2.18}$$

Thus $P_0(N)$ is true iff N is inaccessible, $P_1(N)$ is true iff N is the Nth inaccessible (which we have seen to be true for Mahlo cardinals), $P_2(N)$ is true iff N is the Nth cardinal having property P_1 (which we have seen to be true for measurable cardinals), etc. So the axiom

$$\big(\forall x \mid \mathsf{Ord}(x) \to \big(\exists N \mid P_x(N)\big)\big)$$

implies the existence of many and very large cardinals. And, if one likes, one can repeat this construction after replacing the predicate '$\mathsf{Is\_inaccessible}$' in (2.18) by

$$\big(\exists K \,\big|\, \big(\forall x \in K \mid \mathsf{Ord}(x) \to \big(\exists N \mid P_x(N)\big)\big)\big).$$

These particular statements do not seem to have been studied enough for surmises concerning their consistency or inconsistency to have developed. But if they are all consistent, there will exist *inner models* of set theory, in the sense described in the next section, in which any finite collection of them are true. This will allow theories containing such axioms to be covered by 'axioms of reflection' of the kind that will be discussed in Sect. 6.3. Of course, all of this resembles the play of children with large numbers: *'a thousand trillion gazillion **plus one**'*.

2.4.2.4 More General 'Inner' Models of Set Theory

A predicate model of the Zermelo–Fraenkel axioms must provide some set $\mathscr{U}$ as universe and assign a two-variable Boolean function E on $\mathscr{U}$ to represent the non-logical symbol '$\in$'. The most direct (but of course not the only) way of doing this is to choose a set $\mathscr{U}$ having appropriate properties and simply to define E as

$$E(x, y) = \textbf{if } x \in y \textbf{ then } 1 \textbf{ else } 0 \textbf{ end if},$$

which can be written more simply as

$$E(x, y) \leftrightarrow (x \in y)$$

if we agree to represent predicates by true/false-valued, rather than 0/1-valued, functions. (An element $A(x)$ of $\mathscr{U}$ must be assigned to each free variable x appearing in a term or formula whose value is to be calculated.) Using this convention, and noting that the ZFC axioms involve no function symbols and so they do not require formation of any terms, we can write our previous recursive rules for calculating the value associated with each predicate expression F (cf. Sect. 2.2) in the following slightly specialized way:

(i) If the expression F is just an individual variable x, then $\mathsf{Val}(A, F) = A(x)$.
(ii) If F is an atomic formula having the form '$x \in y$', then $\mathsf{Val}(A, F)$ is the Boolean value $A(x) \in A(y)$.
(iii) If F is a formula having the form $(\forall v_1, \ldots, v_k \mid e)$, then $\mathsf{Val}(A, F)$ is

$$\big(\forall x_1, \ldots, x_k \mid (x_1 \in \mathscr{U} \,\&\, \cdots \,\&\, x_k \in \mathscr{U}) \rightarrow \mathsf{Val}\big(A(x_1, \ldots, x_k), e\big)\big),$$

where $A(x_1, \ldots, x_k)$ assigns the same value as A to every free variable of e, but assigns the value x_j to each v_j, for j from 1 to k.
(iv) If F is a formula having the form $(\exists v_1, \ldots, v_k \mid e)$, then $\mathsf{Val}(A, F)$ is

$$\big(\exists x_1, \ldots, x_k \mid (x_1 \in \mathscr{U} \,\&\, \cdots \,\&\, x_k \in \mathscr{U}) \,\&\, \mathsf{Val}\big(A(x_1, \ldots, x_k), e\big)\big),$$

where $A(x_1, \ldots, x_k)$ assigns the same value as A to every free variable of e, but assigns the value x_j to each v_j, for j from 1 to k.
(v) If the formula F has the form '$G \,\&\, H$', then $\mathsf{Val}(A, F)$ is $\mathsf{Val}(A, G) \,\&\, \mathsf{Val}(A, H)$.
(vi) If the formula F has the form '$G \vee H$', then $\mathsf{Val}(A, F)$ is $\mathsf{Val}(A, G) \vee \mathsf{Val}(A, H)$.
(vii) If the formula F has the form '$\neg G$', then

$$\mathsf{Val}(A, F) = \big(\neg \mathsf{Val}(A, G)\big).$$

(viii) If the formula F has the form '$G \rightarrow H$', then $\mathsf{Val}(A, F)$ is

$$\mathsf{Val}(A, G) \rightarrow \mathsf{Val}(A, H).$$

(ix) If the formula F has the form '$G \leftrightarrow H$', then $\mathsf{Val}(A, F)$ is

$$\mathsf{Val}(A, G) \leftrightarrow \mathsf{Val}(A, H).$$

The set $\mathscr{U}$ defines a model of ZFC if and only if each of the ZFC axioms evaluates to 'true' under these rules. We shall pinpoint in Sect. 6.3 conditions on $\mathscr{U}$ sufficient for this to be the case.

We will generally suppose that $\mathscr{U}$ is *transitive*, i.e. that each member of a member of $\mathscr{U}$ is also a member of $\mathscr{U}$. Then axiom (1) of ZFC evaluates to

$$\big(\forall s, t \mid (s \in \mathscr{U}\ \&\ t \in \mathscr{U}) \rightarrow \big(s = t \leftrightarrow \big(\forall x \mid (x \in \mathscr{U}) \rightarrow \big((x \in s) \leftrightarrow (x \in t)\big)\big)\big)\big).$$

This formula clearly has the value true. Indeed, if $s = t$, then $(x \in s) \leftrightarrow (x \in t)$ for every $x \in \mathscr{U}$, so clearly

$$\big(\forall x \mid (x \in \mathscr{U}) \rightarrow \big((x \in s) \leftrightarrow (x \in t)\big)\big) \tag{2.19}$$

must be true. Suppose conversely that $s \neq t$. Then by the ZFC axiom of extensionality, one of these sets, say s, has a member x that is not in the other. Since $\mathscr{U}$ is transitive we have $x \in \mathscr{U}$, so (2.19) must be false.

ZFC axiom (6) (axiom of regularity) evaluates to

$$\neg\big(\exists x \mid (x \in \mathscr{U})\ \&\ (x \neq \emptyset)$$
$$\&\ \big(\forall y \,\big|\, \big((y \in \mathscr{U})\ \&\ (y \in x)\big) \rightarrow \big(\exists z \mid (z \in \mathscr{U})\ \&\ (z \in x)\ \&\ (z \in y)\big)\big)\big),$$

and this also must be true. Indeed, if x in $\mathscr{U}$ is non-null, then by the ZFC axiom of regularity it must have an element y which is disjoint from it, and since $\mathscr{U}$ is transitive this y is also in $\mathscr{U}$.

References

[Jec97] Jech, T.J.: Set Theory, 2nd edn. Perspectives in Mathematical Logic. Springer, Berlin (1997)

http://www.springer.com/978-0-85729-807-2

Computational Logic and Set Theory
Applying Formalized Logic to Analysis
Schwartz, J.T.; Cantone, D.; Omodeo, E.G.
2011, XVII, 416 p., Hardcover
ISBN: 978-0-85729-807-2