1	Ω	N	PA	$A \times B$
$*$	$a = a'$	0	$\{x \in A \| \phi(x)\}$	$< a, b >$
	$a \in \alpha$	Sn		

— it being assumed that a and a' are terms of type A already constructed, α of type PA, n of type N, $\phi(x)$ of type Ω and b of type B.

Logical symbols may be defined as follows:

$$\begin{aligned}
\top &\equiv * = *, \\
p \wedge q &\equiv < p, q > = < \top, \top >, \\
p \Rightarrow q &\equiv p \wedge q = p, \\
\forall_{x \in A} \phi(x) &\equiv \{x \in A \mid \phi(x)\} = \{x \in A \mid \top\},
\end{aligned}$$

where it is understood that p, q and $\phi(x)$ are terms of type Ω. From these symbols one may define others, taking care not to make implicit use of De Morgan's rules (Prawitz [1965]):

$$\begin{aligned}
\bot &\equiv \forall_{t \in \Omega} t, \\
\neg p &\equiv \forall_{t \in \Omega} (p \Rightarrow t), \\
p \vee q &\equiv \forall_{t \in \Omega} (((p \Rightarrow t) \wedge (q \Rightarrow t)) \Rightarrow t), \\
\exists_{x \in A} \phi(x) &\equiv \forall_{t \in \Omega} ((\forall_{x \in a} (\phi(x) \Rightarrow t)) \Rightarrow t).
\end{aligned}$$

Other symbols, such as appear in $\exists_{x \in A} \phi(x)$, $\{a\}$, $\alpha \subseteq \beta$, $\alpha \times \beta$, etc., are defined in the usual fashion.

Axioms and rules of inference are stated in terms of a deduction symbol $\vdash_X$, where X is a finite set of variables. The permissible deductions take the form

$$p_1, \ldots, p_n \vdash_X p_{n+1},$$

where the p_i are terms of type Ω and X contains all the variables which occur freely in the p_i. (When X is empty, the subscript may be ommitted.) The axioms and rules of inference hold no surprises. For the purpose of illustration, here are a few special cases:

- $p \vdash p$;
- $\dfrac{\phi(x) \vdash_{\{x\}} \psi(x)}{\phi(\alpha) \vdash \psi(\alpha)}$;
- $< a, b > = < c, d > \vdash a = c$;
- $\dfrac{\phi(x) \vdash_{\{x\}} \phi(Sx)}{\phi(0) \vdash_{\{x\}} \phi(x)}$.

The reader may recognize the last mentioned rule of inference as the principle of mathematical induction. Although we have not stated all the axioms and rules of inference here, they will of course imply the usual axioms and rules for the intuitionistic propositional and predicate calculi.

For a less formal treatment of constructive arithmetic and analysis, the reader may consult Goodstein [1970] and Bishop [1967], respectively.

28

Gödel's Theorems

A formal language is (among other things) a system for dealing with strings of symbols. An interpretation of these symbols is called a *model.* For example, the Lambda Calculus tells us how to arrange marks such as ' and λ_x. The interpretation of these marks in terms of functional application and functional abstraction is a model.

The formal languages that interest us here contain the notion of a proof. A *proof* is a finite sequence of formulas, each of which is either an axiom or follows from some previous members of the sequence by a rule of inference. We call a formal language *consistent* if there is no proof in the language whose last line is $\perp$, that is, it does not contain the proof of a contradiction.

In this chapter we consider a formal language L which is adequate for *arithmetic.* We assume that L includes the intuitionistic predicate calculus together with some axioms for arithmetic, and we assume that all the basic laws of arithmetic are provable in L. For example, L might be the type theory considered in the previous chapter.

In 1930, Kurt Gödel (1906–1978) proved a completeness result for the classical predicate calculus. In 1950, this result was extended by Leon Henkin to classical type theory. It was later extended to intuitionistic type theory (Lambek and Scott [1986]). This completeness result may be expressed as follows:

Theorem 28.1. (Completeness)
A formula is provable in L if it is true under all possible interpretations of the nonlogical symbols in L, i.e., in all models of L.