

Contents

Introduction	XIII
Notation	XVI
1 Propositional Logic	1
1.1 Boolean Functions and Formulas	2
1.2 Semantic Equivalence and Normal Forms	9
1.3 Tautologies and Logical Consequence	14
1.4 A Complete Calculus for $\models$	18
1.5 Applications of the Compactness Theorem	25
1.6 Hilbert Calculi	29
2 Predicate Logic	33
2.1 Mathematical Structures	34
2.2 Syntax of Elementary Languages	43
2.3 Semantics of Elementary Languages	49
2.4 General Validity and Logical Equivalence	58
2.5 Logical Consequence and Theories	62
2.6 Explicit Definitions—Expanding Languages	67
3 Gödel's Completeness Theorem	71
3.1 A Calculus of Natural Deduction	72
3.2 The Completeness Proof	76
3.3 First Applications—Nonstandard Models	81
3.4 ZFC and Skolem's Paradox	87
3.5 Enumerability and Decidability	92
3.6 Complete Hilbert Calculi	95
3.7 First-Order Fragments and Extensions	99

4	The Foundations of Logic Programming	105
4.1	Term Models and Horn Formulas	106
4.2	Propositional Resolution	112
4.3	Unification	119
4.4	Logic Programming	122
4.5	Proof of the Main Theorem	129
5	Elements of Model Theory	131
5.1	Elementary Extensions	132
5.2	Complete and κ -Categorical Theories	137
5.3	Ehrenfeucht's game	142
5.4	Embedding and Characterization Theorems	145
5.5	Model Completeness	151
5.6	Quantifier Elimination	157
5.7	Reduced Products and Ultraproducts	163
6	Incompleteness and Undecidability	167
6.1	Recursive and Primitive Recursive Functions	169
6.2	Arithmetization	176
6.3	Representability of Arithmetical Predicates	182
6.4	The Representability Theorem	189
6.5	The Theorems of Gödel, Tarski, Church	194
6.6	Transfer by Interpretation	200
6.7	The Arithmetical Hierarchy	205
7	On the Theory of Self-Reference	209
7.1	The Derivability Conditions	210
7.2	The Theorems of Gödel and Löb	217
7.3	The Provability Logic G	221
7.4	The Modal Treatment of Self-Reference	223
7.5	A Bimodal Provability Logic for PA	226
7.6	Modal Operators in ZFC	228
	Hints to the Exercises	231
	Literature	241

Index of Terms and Names	247
---------------------------------	------------

Index of Symbols	255
-------------------------	------------

Traditional logic as a part of philosophy is one of the oldest scientific disciplines. It can be traced back to the Stoics and to Aristotle.¹ It is one of the roots of what is nowadays called philosophical logic. Mathematical logic, however, is a relatively young discipline, having arisen from the endeavors of Peano, Frege and Russell to reduce mathematics entirely to logic. It steadily developed during the twentieth century into a broad discipline with several subareas and numerous applications in mathematics, computer science, linguistics and philosophy.

One of the features of modern logic is a clear separation between object language and metalanguage. The latter is necessarily a kind of a *privileged language*, although it differs from author to author and depends also on the context. The author has in mind, in any case, it is mixed up with *semantical* concepts, as it already have their origin in set theory. The amount of set theory involved depends on one's objectives. General semantics and model theory use stronger set-theoretical tools than does proof theory. But on average, little more is assumed than knowledge of the most common set-theoretical terminology presented in almost every mathematical course for beginners. Much of it is used only as a *fagon de parler*.

Since this book concerns mathematical logic, its language is similar to the language common to all mathematical disciplines. There is one essential difference though. In mathematics, metalanguage and object language strongly interact with each other and the latter is semi-formalized in the best of cases. This method has proved successful. Separating object language and metalanguage is relevant only in special context, for example in axiomatic set theory where formalisation is needed to specify how certain axioms look like. Strictly formal languages are met more often in computer science. In analyzing complex software or a programming language like in logic, formal linguistic entities are the objects of consideration.

The way of arguing about formal languages and theories is traditionally called the *metatheory*. An important task of a metatheoretical analysis is to specify procedures of logical inference by so-called *logical calculi*, which operate purely syntactically. There are many different logical calculi. The choice may depend on the formalized language, on the logical basis, and on certain aims of the formalization. Basic metatheoretical tools are in any case the naive natural numbers and inductive proof procedures. We will sometimes call them proofs by *metainduction*, in particular when talking about formalized theories that may speak about natural numbers and induction themselves. Induction can likewise be carried out on certain sets of strings over a fixed alphabet, or on the system of rules of a logical calculus.

¹The Aristotelian syllogisms are useful examples for inferences in a first-order language with unary predicate symbols. One of these serves as an example in Section 4.4 on logic programming.