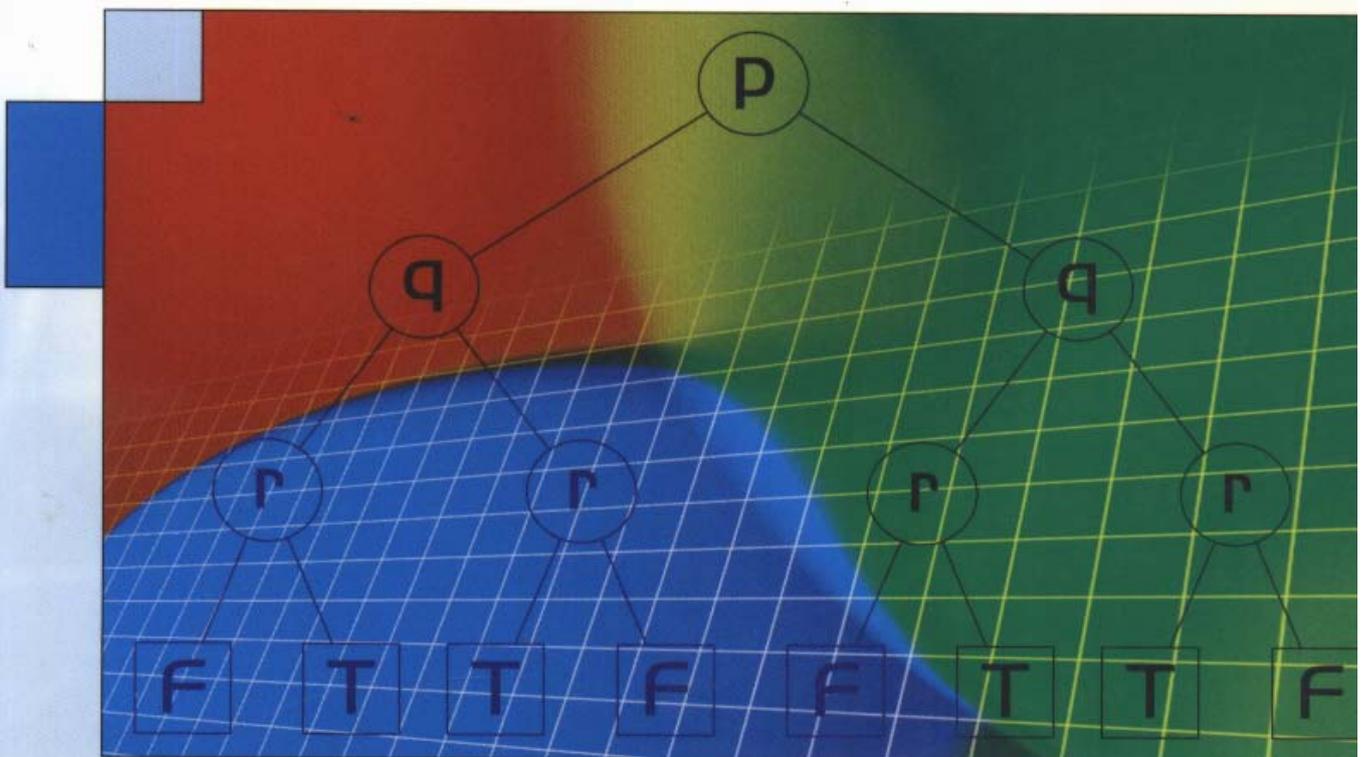


Mordechai Ben-Ari

Mathematical Logic *for* Computer Science

[Second edition]



Springer

Contents

Preface	vii
1 Introduction	1
1.1 The origins of mathematical logic	1
1.2 Propositional calculus	2
1.3 Predicate calculus	3
1.4 Theorem proving and logic programming	5
1.5 Systems of logic	6
1.6 Exercise	7
2 Propositional Calculus: Formulas, Models, Tableaux	9
2.1 Boolean operators	9
2.2 Propositional formulas	12
2.3 Interpretations	17
2.4 Logical equivalence and substitution	19
2.5 Satisfiability, validity and consequence	24
2.6 Semantic tableaux	29
2.7 Soundness and completeness	33
2.8 Implementation ^P	38
2.9 Exercises	40
3 Propositional Calculus: Deductive Systems	43
3.1 Deductive proofs	43
3.2 The Gentzen system \mathcal{G}	45
3.3 The Hilbert system \mathcal{H}	48
3.4 Soundness and completeness of \mathcal{H}	56
3.5 A proof checker ^P	59
3.6 Variant forms of the deductive systems*	60
3.7 Exercises	64

4	Propositional Calculus: Resolution and BDDs	67
4.1	Resolution	67
4.2	Binary decision diagrams (BDDs)	81
4.3	Algorithms on BDDs	88
4.4	Complexity*	95
4.5	Exercises	99
5	Predicate Calculus: Formulas, Models, Tableaux	101
5.1	Relations and predicates	101
5.2	Predicate formulas	102
5.3	Interpretations	105
5.4	Logical equivalence and substitution	107
5.5	Semantic tableaux	109
5.6	Implementation ^P	118
5.7	Finite and infinite models*	120
5.8	Decidability*	121
5.9	Exercises	125
6	Predicate Calculus: Deductive Systems	127
6.1	The Gentzen system \mathcal{G}	127
6.2	The Hilbert system \mathcal{H}	129
6.3	Implementation ^P	134
6.4	Complete and decidable theories*	135
6.5	Exercises	138
7	Predicate Calculus: Resolution	139
7.1	Functions and terms	139
7.2	Clausal form	142
7.3	Herbrand models	148
7.4	Herbrand's Theorem*	150
7.5	Ground resolution	152
7.6	Substitution	153
7.7	Unification	155
7.8	General resolution	164

7.9	Exercises	171
8	Logic Programming	173
8.1	Formulas as programs	173
8.2	SLD-resolution	176
8.3	Prolog	181
8.4	Concurrent logic programming*	186
8.5	Constraint logic programming*	194
8.6	Exercises	199
9	Programs: Semantics and Verification	201
9.1	Introduction	201
9.2	Semantics of programming languages	202
9.3	The deductive system \mathcal{HL}	209
9.4	Program verification	211
9.5	Program synthesis	213
9.6	Soundness and completeness of \mathcal{HL}	216
9.7	Exercises	219
10	Programs: Formal Specification with \mathcal{Z}	221
10.1	Case study: a traffic signal	221
10.2	The \mathcal{Z} notation	224
10.3	Case study: semantic tableaux	230
10.4	Exercises	234
11	Temporal Logic: Formulas, Models, Tableaux	235
11.1	Introduction	235
11.2	Syntax and semantics	236
11.3	Models of time	239
11.4	Semantic tableaux	242
11.5	Implementation of semantic tableaux ^P	252
11.6	Exercises	255
12	Temporal Logic: Deduction and Applications	257
12.1	The deductive system \mathcal{L}	257

12.2	Soundness and completeness of \mathcal{L}^*	262
12.3	Other temporal logics*	264
12.4	Specification and verification of programs*	266
12.5	Model checking*	272
12.6	Exercises	280
A	Set Theory	283
A.1	Finite and infinite sets	283
A.2	Set operators	284
A.3	Ordered sets	286
A.4	Relations and functions	287
A.5	Cardinality	288
A.6	Proving properties of sets	289
B	Further Reading	291
	Bibliography	293
	Index of Symbols	297
	Index	299

Mathematical Logic for Computer Science is a mathematics textbook with theorems and proofs, but the choice of topics has been guided by the needs of computer science students. The method of semantic tableaux provides an elegant way to teach logic that is both theoretically sound and yet sufficiently elementary for undergraduates. To provide a balanced treatment of logic, tableaux are related to deductive proof systems.

The logical systems presented are:

- Propositional calculus (including binary decision diagrams)
- Predicate calculus
- Resolution
- Hoare logic
- λ
- Temporal logic

Answers to exercises as well as Prolog source code for algorithms are available via the Springer London website:

http://www.springer.co.uk/com_pub/ct_mlcs.html

Mordechai Ben-Ari is Associate Professor at the Department of Science Teaching of the Weizmann Institute of Science. He has published textbooks on concurrent programming and programming languages.

ISBN 1-85233-319-7



<http://www.springer.co.uk>