

Contents

<i>Foreword to the first edition</i>	page ix
<i>Preface to the second edition</i>	xi
<i>Acknowledgements</i>	xiii
1 Propositional logic	1
1.1 Declarative sentences	2
1.2 Natural deduction	5
1.2.1 Rules for natural deduction	6
1.2.2 Derived rules	23
1.2.3 Natural deduction in summary	26
1.2.4 Provable equivalence	29
1.2.5 An aside: proof by contradiction	29
1.3 Propositional logic as a formal language	31
1.4 Semantics of propositional logic	36
1.4.1 The meaning of logical connectives	36
1.4.2 Mathematical induction	40
1.4.3 Soundness of propositional logic	45
1.4.4 Completeness of propositional logic	49
1.5 Normal forms	53
1.5.1 Semantic equivalence, satisfiability and validity	54
1.5.2 Conjunctive normal forms and validity	58
1.5.3 Horn clauses and satisfiability	65
1.6 SAT solvers	68
1.6.1 A linear solver	69
1.6.2 A cubic solver	72
1.7 Exercises	78
1.8 Bibliographic notes	91
2 Predicate logic	93
2.1 The need for a richer language	93

2.2	Predicate logic as a formal language	98
2.2.1	Terms	99
2.2.2	Formulas	100
2.2.3	Free and bound variables	102
2.2.4	Substitution	104
2.3	Proof theory of predicate logic	107
2.3.1	Natural deduction rules	107
2.3.2	Quantifier equivalences	117
2.4	Semantics of predicate logic	122
2.4.1	Models	123
2.4.2	Semantic entailment	129
2.4.3	The semantics of equality	130
2.5	Undecidability of predicate logic	131
2.6	Expressiveness of predicate logic	136
2.6.1	Existential second-order logic	139
2.6.2	Universal second-order logic	140
2.7	Micromodels of software	141
2.7.1	State machines	142
2.7.2	Alma – re-visited	146
2.7.3	A software micromodel	148
2.8	Exercises	157
2.9	Bibliographic notes	170
3	Verification by model checking	172
3.1	Motivation for verification	172
3.2	Linear-time temporal logic	175
3.2.1	Syntax of <i>LTL</i>	175
3.2.2	Semantics of <i>LTL</i>	178
3.2.3	Practical patterns of specifications	183
3.2.4	Important equivalences between <i>LTL</i> formulas	184
3.2.5	Adequate sets of connectives for <i>LTL</i>	186
3.3	Model checking: systems, tools, properties	187
3.3.1	Example: mutual exclusion	187
3.3.2	The NuSMV model checker	191
3.3.3	Running NuSMV	194
3.3.4	Mutual exclusion revisited	195
3.3.5	The ferryman	199
3.3.6	The alternating bit protocol	203
3.4	Branching-time logic	207
3.4.1	Syntax of <i>CTL</i>	208

3.4.2	Semantics of CTL	211
3.4.3	Practical patterns of specifications	215
3.4.4	Important equivalences between CTL formulas	215
3.4.5	Adequate sets of CTL connectives	216
3.5	CTL* and the expressive powers of LTL and CTL	217
3.5.1	Boolean combinations of temporal formulas in CTL	220
3.5.2	Past operators in LTL	221
3.6	Model-checking algorithms	221
3.6.1	The CTL model-checking algorithm	222
3.6.2	CTL model checking with fairness	230
3.6.3	The LTL model-checking algorithm	232
3.7	The fixed-point characterisation of CTL	238
3.7.1	Monotone functions	240
3.7.2	The correctness of SAT_{EG}	242
3.7.3	The correctness of SAT_{EU}	243
3.8	Exercises	245
3.9	Bibliographic notes	254
4	Program verification	256
4.1	Why should we specify and verify code?	257
4.2	A framework for software verification	258
4.2.1	A core programming language	259
4.2.2	Hoare triples	262
4.2.3	Partial and total correctness	265
4.2.4	Program variables and logical variables	268
4.3	Proof calculus for partial correctness	269
4.3.1	Proof rules	269
4.3.2	Proof tableaux	273
4.3.3	A case study: minimal-sum section	287
4.4	Proof calculus for total correctness	292
4.5	Programming by contract	296
4.6	Exercises	299
4.7	Bibliographic notes	304
5	Modal logics and agents	306
5.1	Modes of truth	306
5.2	Basic modal logic	307
5.2.1	Syntax	307
5.2.2	Semantics	308
5.3	Logic engineering	316
5.3.1	The stock of valid formulas	317

5.3.2	Important properties of the accessibility relation	320
5.3.3	Correspondence theory	322
5.3.4	Some modal logics	326
5.4	Natural deduction	328
5.5	Reasoning about knowledge in a multi-agent system	331
5.5.1	Some examples	332
5.5.2	The modal logic $KT45^n$	335
5.5.3	Natural deduction for $KT45^n$	339
5.5.4	Formalising the examples	342
5.6	Exercises	350
5.7	Bibliographic notes	356
6	Binary decision diagrams	358
6.1	Representing boolean functions	358
6.1.1	Propositional formulas and truth tables	359
6.1.2	Binary decision diagrams	361
6.1.3	Ordered BDDs	366
6.2	Algorithms for reduced OBDDs	372
6.2.1	The algorithm <code>reduce</code>	372
6.2.2	The algorithm <code>apply</code>	373
6.2.3	The algorithm <code>restrict</code>	377
6.2.4	The algorithm <code>exists</code>	377
6.2.5	Assessment of OBDDs	380
6.3	Symbolic model checking	382
6.3.1	Representing subsets of the set of states	383
6.3.2	Representing the transition relation	385
6.3.3	Implementing the functions <code>pre\exists</code> and <code>pre\forall</code>	387
6.3.4	Synthesising OBDDs	387
6.4	A relational mu-calculus	390
6.4.1	Syntax and semantics	390
6.4.2	Coding CTL models and specifications	393
6.5	Exercises	398
6.6	Bibliographic notes	413
	<i>Bibliography</i>	414
	<i>Index</i>	418