Contents

1. **Introduction** 1
1.1 Formal Methods 2
1.2 Developing and Acquiring Formal Methods 5
1.3 Using Formal Methods 7
1.4 Applying Formal Methods 9
1.5 Overview of the Book 11

2. **Preliminaries** 13
2.1 Set Notation 13
2.2 Strings and Languages 15
2.3 Graphs 16
2.4 Computational Complexity and Computability 20
2.5 Further Reading 27

3. **Logic and Theorem Proving** 29
3.1 First Order Logic 29
3.2 Terms 30
3.3 First Order Formulas 33
3.4 Propositional Logic 39
3.5 Proving First Order Logic Formulas 39
3.6 Properties of Proof Systems 43
3.7 Proving Propositional Logic Properties 46
3.8 A Practical Proof System 47
3.9 Example Proofs 50
3.10 Machine Assisted Proofs 59
3.11 Mechanized Theorem Provers 61
3.12 Further Reading 61

4. **Modeling Software Systems** 63
4.1 Sequential, Concurrent and Reactive Systems 64
4.2 States 67
4.3 State Spaces 68
4.4 Transition Systems 71
4.5 Granularity of Transitions 75

4.6 Examples of Modeling Programs 77
4.7 Nondeterministic Transitions 85
4.8 Assigning Propositional Variables to States 86
4.9 Combining State Spaces 88
4.10 The Linear View 90
4.11 The Branching View 91
4.12 Fairness 92
4.13 The Partial Order View 98
4.14 Modeling Formalisms 107
4.15 A Modeling Project 109
4.16 Further Reading 110

5. Formal Specification 113
5.1 Properties of Specification Formalisms 114
5.2 Linear Temporal Logic 116
5.3 Axiomatizing LTL 121
5.4 Examples of LTL Specification 123
5.5 Automata on Infinite Words 127
5.6 Specification using Büchi-automata 129
5.7 Deterministic Büchi Automata 132
5.8 Alternative Specification Formalisms 133
5.9 Complicated Specifications 136
5.10 Completeness of Specification 136
5.11 Further Reading 138

6. Automatic Verification 139
6.1 State Space Search 140
6.2 Representing States 143
6.3 The Automata Framework 143
6.4 Combining Büchi Automata 145
6.5 Complementing a Büchi Automaton 151
6.6 Checking Emptiness 152
6.7 A Model Checking Example 154
6.8 Translating LTL into Automata 156
6.9 The Complexity of Model Checking 164
6.10 Representing Fairness 169
6.11 Checking the LTL Specifications 170
6.12 Safety Properties 171
6.13 The State Space Explosion Problem 172
6.14 Advantages of Model Checking 174
6.15 Weaknesses of Model Checking 174
6.16 Selecting Automatic Verification Tools 175
6.17 Model Checking Projects 175
6.18 Model Checking Tools 176
6.19 Further Reading 177

7. Deductive Software Verification 179
7.1 Verification of Flow Chart Programs 180
7.2 Verification with Array Variables 187
7.3 Total Correctness 190
7.4 Axiomatic Program Verification 195
7.5 Verification of Concurrent Programs 202
7.6 Advantages of Deductive Verification 207
7.7 Weaknesses of Deductive verification 208
7.8 Soundness and Completeness of Proof Systems 210
7.9 Compositionality 212
7.10 Deductive Verification Tools 213
7.11 Further Reading 213

8. Process Algebra and Equivalences 215
8.1 Process Algebras 217
8.2 A Calculus of Communicating Systems 218
8.3 An Example: Dekker's Algorithm 225
8.4 Modeling Issues 229
8.5 Equivalences between Agents 230
8.5.1 Trace equivalence 231
8.5.2 Failure equivalence 232
8.5.3 Simulation Equivalence 233
8.5.4 Bisimulation and Weak Bisimulation equivalence 235
8.6 A Hierarchy of Equivalence Relations 237
8.7 Studying Concurrency using Process Algebra 238
8.8 Calculating Bisimulation Equivalence 242
8.9 LOTOS 245
8.10 Process Algebra Tools 247
8.11 Further Reading 247

9. Software Testing 249
9.1 Inspections and Walkthroughs 251
9.2 Control Flow Coverage Criteria 253
9.3 Dataflow Coverage Criteria 259
9.4 Propagating path conditions 261
9.5 Equivalence Partition 267
9.6 Preparing the Code for Testing 267
9.7 Checking the Test Suite 269
9.8 Compositionality 270
9.9 Black Box Testing 272
9.10 Probabilistic Testing 275
9.11 Advantages of Testing 276
9.12 Disadvantages of Testing 277
9.13 Testing Tools 278
9.14 Further Reading 278

10. Combining Formal Methods 279
10.1 Abstraction 279
10.2 Combining Testing and Model Checking 286
10.2.1 Direct Checking 286
10.2.2 Black Box Systems 287
10.2.3 Combination Lock Automata 288
10.2.4 Black Box Deadlock Detection 289
10.2.5 Conformance Testing 290
10.2.6 Checking the Reliability of Resets 294
10.2.7 Black Box Checking 294
10.3 The Cleanroom Method 297

11. Visualization 299
11.1 Using Visualizations for Formal Methods 300
11.2 Message Sequence Charts 300
11.3 Visualizing Flowcharts and State Machines 305
11.4 Hierarchical State Graphs 308
11.5 Visualizing Program Text 312
11.6 Petri Nets 312
11.7 Visualization Tools 314
11.8 Further Reading 316

12. Conclusions 317

References 321

Index 329