- **1 Elementary Notions and Notations** (p. 1)
- **1.1 A Proof Primer** (p. 2)
- **1.1.1 Logical Statements** (p. 2)
- **1.1.2 Something to Talk About** (p. 5)
- **1.1.3 Proof Techniques** (p. 6)
- **Exercises** (p. 12)
- **1.2 Sets** (p. 13)
- **1.2.1 Definition of a Set** (p. 13)
- **1.2.2 Operations on Sets** (p. 18)
- **1.2.3 Counting Finite Sets** (p. 26)
- **1.2.4 Bags (Multisets)** (p. 29)
- **1.2.5 Sets Should Not Be Too Complicated** (p. 30)
- **Exercises** (p. 31)
- **1.3 Ordered Structures** (p. 35)
- **1.3.1 Tuples** (p. 35)
- **1.3.2 Lists** (p. 39)
- **1.3.3 Strings and Languages** (p. 41)
- **1.3.4 Relations** (p. 46)
- **1.3.5 Counting Tuples** (p. 49)
- **Exercises** (p. 52)
- **1.4 Graphs and Trees** (p. 55)
- **1.4.1 Definition of a Graph** (p. 55)
- **1.4.2 Paths and Graphs** (p. 59)
- **1.4.3 Graph Traversals** (p. 61)
- **1.4.4 Trees** (p. 63)
- **1.4.5 Spanning Trees** (p. 68)
- **Exercises** (p. 70)
- **1.5 Chapter Summary** (p. 72)
- **2 Facts about Functions** (p. 73)
- **2.1 Definitions and Examples** (p. 74)
- **2.1.1 Definition of a Function** (p. 74)
- **2.1.2 Some Useful Functions** (p. 79)
- **2.1.3 Partial Functions** (p. 87)
- **Exercises** (p. 88)
- **2.2 Constructing Functions** (p. 91)
- **2.2.1 Composition of Functions** (p. 91)
- **2.2.2 The Map Function** (p. 96)
- **Exercise** (p. 98)
- **2.3 Properties of Functions** (p. 100)
- **2.3.1 Injections and Surjections** (p. 100)
- **2.3.2 Bijections and Inverses** (p. 102)
- **2.3.3 The Pigeonhole Principle** (p. 105)
- **2.3.4 Simple Ciphers** (p. 106)
- **2.3.5 Hash Functions** (p. 109)
- **Exercises** (p. 111)
- **2.4 Countability** (p. 115)

- **2.4.1 Comparing the Size of Sets** (p. 115)
- **2.4.2 Sets that Are Countable** (p. 116)
- **2.4.3 Diagonalization** (p. 119)
- **2.4.4 Limits on Computability** (p. 121)
- **Exercises** (p. 124)
- **2.5 Chapter Summary** (p. 125)
- **3 Construction Techniques** (p. 127)
- **3.1 Inductively Defined Sets** (p. 128)
- **3.1.1 Numbers** (p. 129)
- **3.1.2 Strings** (p. 132)
- **3.1.3 Lists** (p. 134)
- **3.1.4 Binary Trees** (p. 138)
- **3.1.5 Cartesian Products of Sets** (p. 140)
- **Exercises** (p. 142)
- **3.2 Recursive Functions and Procedures** (p. 145)
- **3.2.1 Numbers** (p. 146)
- **3.2.2 Strings** (p. 150)
- **3.2.3 Lists** (p. 153)
- **3.2.4 Binary Trees** (p. 159)
- **3.2.5 Two More Problems** (p. 163)
- **3.2.6 Infinite Sequences** (p. 165)
- **Exercises** (p. 168)
- **3.3 Grammars** (p. 173)
- **3.3.1 Recalling an English Grammar** (p. 173)
- **3.3.2 Structure of Grammars** (p. 174)
- **3.3.3 Derivations** (p. 177)
- **3.3.4 Constructing Grammars** (p. 181)
- **3.3.5 Meaning and Ambiguity** (p. 186)
- **Exercises** (p. 188)
- **3.4 Chapter Summary** (p. 191)
- **4 Equivalence, Order, and Inductive Proof** (p. 193)
- **4.1 Properties of Binary Relations** (p. 194)
- **4.1.1 Composition of Relations** (p. 195)
- **4.1.2 Closures** (p. 199)
- **4.1.3 Path Problems** (p. 204)
- **Exercises** (p. 209)
- **4.2 Equivalence Relations** (p. 213)
- **4.2.1 Definition and Examples** (p. 214)
- **4.2.2 Equivalence Classes** (p. 218)
- **4.2.3 Partitions** (p. 219)
- **4.2.4 Generating Equivalence Relations** (p. 225)
- **Exercises** (p. 229)
- **4.3 Order Relations** (p. 232)
- **4.3.1 Partial Orders** (p. 233)
- **4.3.2 Topological Sorting** (p. 239)
- **4.3.3 Well-Founded Orders** (p. 242)

- **4.3.4 Ordinal Numbers** (p. 250)
- **Exercises** (p. 251)
- **4.4 Inductive Proof** (p. 253)
- **4.4.1 Proof by Mathematical Induction** (p. 253)
- **4.4.2 Proof by Well-Founded Induction** (p. 259)
- **4.4.3 A Variety of Examples** (p. 261)
- **Exercises** (p. 267)
- **4.5 Chapter Summary** (p. 272)
- **5 Analysis Techniques** (p. 273)
- **5.1 Analyzing Algorithms** (p. 274)
- **5.1.1 Worst-Case Running Time** (p. 274)
- **5.1.2 Decision Trees** (p. 277)
- **Exercises** (p. 281)
- **5.2 Finding Closed Forms** (p. 281)
- **5.2.1 Closed Forms for Sums** (p. 282)
- **Exercises** (p. 287)
- **5.3 Counting and Discrete Probability** (p. 289)
- **5.3.1 Permutations (Order Is Important)** (p. 289)
- **5.3.2 Combinations (Order Is Not Important)** (p. 293)
- **5.3.3 Discrete Probability** (p. 298)
- **Exercises** (p. 309)
- **5.4 Solving Recurrences** (p. 312)
- **5.4.1 Solving Simple Recurrences** (p. 313)
- **5.4.2 Generating Functions** (p. 319)
- **Exercises** (p. 332)
- **5.5 Comparing Rates of Growth** (p. 334)
- **5.5.1 Big Theta** (p. 334)
- **5.5.2 Little Oh** (p. 338)
- **5.5.3 Big Oh and Big Omega** (p. 339)
- **Exercises** (p. 341)
- **6.1 How Do We Reason?** (p. 346)
- **5.6 Chapter Summary** (p. 342)
- **6 Elementary Logic** (p. 345)
- **6.1.1 What Is a Calculus?** (p. 347)
- **6.1.2 How Can We Tell Whether Something Is a Proof?** (p. 348)
- **6.2 Propositional Calculus** (p. 348)
- **6.2.1 Well-Formed Formulas and Semantics** (p. 349)
- **6.2.2 Equivalence** (p. 353)
- **6.2.3 Truth Functions and Normal Forms** (p. 358)
- **6.2.4 Complete Sets of Connectives** (p. 365)
- **Exercises** (p. 367)
- **6.3 Formal Reasoning** (p. 369)
- **6.3.1 Inference Rules** (p. 370)
- **6.3.2 Formal Proof** (p. 372)
- **6.3.3 Proof Notes** (p. 380)
- **Exercises** (p. 381)

- **6.4 Formal Axiom Systems** (p. 384)
- **6.4.1 An Example Axiom System** (p. 384)
- **6.4.2 Other Axiom Systems** (p. 391)
- **Exercises** (p. 392)
- **6.5 Chapter Summary** (p. 394)
- **7 Predicate Logic** (p. 397)
- **7.1 First-Order Predicate Calculus** (p. 397)
- **7.1.1 Predicates and Quantifiers** (p. 398)
- **7.1.2 Well-Formed Formulas** (p. 402)
- **7.1.3 Semantics and Interpretations** (p. 404)
- **7.1.4 Validity** (p. 409)
- **7.1.5 The Validity Problem** (p. 413)
- **Exercises** (p. 413)
- **7.2 Equivalent Formulas** (p. 416)
- **7.2.1 Equivalence** (p. 416)
- **7.2.2 Normal Forms** (p. 424)
- **7.2.3 Formalizing English Sentences** (p. 427)
- **7.2.4 Summary** (p. 429)
- **Exercises** (p. 430)
- **7.3 Formal Proofs in Predicate Calculus** (p. 432)
- **7.3.1 Universal Instantiation** (p. 433)
- **7.3.2 Existential Generalization (EG)** (p. 437)
- **7.3.3 Existential Instantiation (EI)** (p. 438)
- **7.3.4 Universal Generalization (UG)** (p. 440)
- **7.3.5 Examples of Formal Proofs** (p. 443)
- **7.3.6 Summary of Quantifier Proofs Rules** (p. 450)
- **Exercises** (p. 451)
- **7.4 Chapter Summary** (p. 456)
- **8 Applied Logic** (p. 457)
- **8.1 Equality** (p. 458)
- **8.1.1 Describing Equality** (p. 458)
- **8.1.2 Extending Equals for Equals** (p. 464)
- **Exercises** (p. 465)
- **8.2 Program Correctness** (p. 466)
- **8.2.1 Imperative Program Correctness** (p. 467)
- **8.2.2 Array Assignment** (p. 478)
- **8.2.3 Termination** (p. 482)
- **Exercises** (p. 486)
- **8.3 Higher-Order Logics** (p. 491)
- **8.3.1 Classifying Higher-Order Logics** (p. 492)
- **8.3.2 Semantics** (p. 496)
- **8.3.3 Higher-Order Reasoning** (p. 498)
- **Exercises** (p. 501)
- **8.4 Chapter Summary** (p. 503)
- **9 Computational Logic** (p. 505)
- **9.1 Automatic Reasoning** (p. 505)

- **9.1.4 Resolution: The General Case** (p. 521)
- **9.1.1 Clauses and Clausal Forms** (p. 506)
- **9.1.2 Resolution for Propositions** (p. 512)
- **9.1.3 Substitution and Unification** (p. 514)
- **9.1.5 Theorem Proving with Resolution** (p. 526)
- **9.1.6 Remarks** (p. 529)
- **Exercises** (p. 530)
- **9.2 Logic Programming** (p. 533)
- **9.2.1 Family Trees** (p. 534)
- **9.2.2 Definition of a Logic Program** (p. 536)
- **9.2.3 Resolution and Logic Programming** (p. 537)
- **9.2.4 Logic Programming Techniques** (p. 549)
- **Exercises** (p. 553)
- **9.3 Chapter Summary** (p. 555)
- **10 Algebraic Structures and Techniques** (p. 557)
- **10.1 What Is an Algebra?** (p. 558)
- **10.1.1 Definition of an Algebra** (p. 560)
- **10.1.2 Concrete Versus Abstract** (p. 562)
- **10.1.3 Working in Algebras** (p. 564)
- **Exercises** (p. 570)
- **10.2 Boolean Algebra** (p. 572)
- **10.2.1 Simplifying Boolean Expressions** (p. 574)
- **10.2.2 Digital Circuits** (p. 578)
- **Exercises** (p. 583)
- **10.3 Abstract Data Types as Algebras** (p. 585)
- **10.3.1 Natural Numbers** (p. 585)
- **10.3.2 Lists and Strings** (p. 589)
- **10.3.3 Stacks and Queues** (p. 592)
- **10.3.4 Binary Trees and Priority Queues** (p. 596)
- **Exercises** (p. 599)
- **10.4 Computational Algebras** (p. 601)
- **10.4.1 Relational Algebras** (p. 601)
- **10.4.2 Functional Algebras** (p. 607)
- **Exercises** (p. 611)
- **10.5 Other Algebraic Ideas** (p. 613)
- **10.5.1 Congruence** (p. 613)
- **10.5.2 Cryptology: The RSA Algorithm** (p. 616)
- **10.5.3 Subalgebras** (p. 621)
- **10.5.4 Morphisms** (p. 623)
- **Exercises** (p. 629)
- **10.6 Chapter Summary** (p. 632)
- **11 Regular Languages and Finite Automata** (p. 633)
- **11.1 Regular Languages** (p. 634)
- **11.1.1 Regular Expressions** (p. 635)
- **11.1.2 The Algebra of Regular Expressions** (p. 638)
- **Exercises** (p. 640)

- **11.2 Finite Automata** (p. 642)
- **11.2.1 Deterministic Finite Automata** (p. 642)
- **11.2.2 Nondeterministic Finite Automata** (p. 646)
- **11.2.3 Transforming Regular Expressions into Finite Automata** (p. 648)
- **11.2.4 Transforming Finite Automata into Regular Expressions** (p. 650)
- **11.2.5 Finite Automata as Output Devices** (p. 655)
- **11.2.6 Representing and Executing Finite Automata** (p. 658)
- **Exercises** (p. 664)
- **11.3 Constructing Efficient Finite Automata** (p. 666)
- **11.3.1 Another Regular Expression to NFA Algorithm** (p. 667)
- **11.3.2 Transforming an NFA into a DFA** (p. 669)
- **11.3.3 Minimum-State DFAs** (p. 675)
- **Exercises** (p. 681)
- **11.4 Regular Language Topics** (p. 683)
- **11.4.1 Regular Grammars** (p. 684)
- **11.4.2 Properties of Regular Languages** (p. 689)
- **Exercises** (p. 693)
- **11.5 Chapter Summary** (p. 695)
- **12 Context-Free Languages and Pushdown Automata** (p. 697)
- **12.1 Context-Free Languages** (p. 697)
- **Exercises** (p. 700)
- **12.2 Pushdown Automata** (p. 700)
- **12.2.1 Equivalent Forms of Acceptance** (p. 703)
- **12.2.2 Context-Free Grammars and Pushdown Automata** (p. 707)
- **12.2.3 Representing and Executing Pushdown Automata** (p. 712)
- **Exercises** (p. 715)
- **12.3 Parsing Techniques** (p. 717)
- **12.3.1 LL(k) Parsing** (p. 717)
- **12.3.2 LR(k) Parsing** (p. 731)
- **Exercises** (p. 744)
- **12.4 Context-Free Language Topics** (p. 746)
- **12.4.1 Transforming Grammars** (p. 746)
- **12.5 Chapter Summary** (p. 756)
- **12.4.2 Properties of Context-Free Languages** (p. 751)
- **Exercises** (p. 755)
- **13 Turing Machines and Equivalent Models** (p. 757)
- **13.1 Turing Machines** (p. 757)
- **13.1.1 Definition of a Turing Machine** (p. 758)
- **13.1.2 Turing Machines with Output** (p. 762)
- **13.1.3 Alternative Definitions** (p. 765)
- **13.1.4 A Universal Turing Machine** (p. 769)
- **Exercises** (p. 773)
- **13.2 The Church-Turing Thesis** (p. 774)
- **13.2.1 Equivalence of Computational Models** (p. 775)
- **13.2.2 A Simple Programming Language** (p. 776)
- **13.2.3 Recursive Functions** (p. 778)

- **13.2.4 Machines That Transform Strings** (p. 781)
- **Exercises** (p. 787)
- **13.3 Chapter Summary** (p. 789)
- **14 Computational Notions** (p. 791)
- **14.1 Computability** (p. 791)
- **14.1.1 Effective Enumerations** (p. 792)
- **14.1.2 The Halting Problem** (p. 795)
- **14.1.3 The Total Problem** (p. 796)
- **14.1.4 Other Problems** (p. 798)
- **Exercises** (p. 802)
- **14.2 A Hierarchy of Languages** (p. 803)
- **14.2.1 The Languages** (p. 803)
- **14.2.2 Summary** (p. 807)
- **Exercises** (p. 807)
- **14.3 Complexity Classes** (p. 808)
- **14.3.1 The Class P** (p. 809)
- **14.3.2 The Class NP** (p. 810)
- **14.3.3 The Class PSPACE** (p. 811)
- **14.3.4 Intractable Problems** (p. 813)
- **14.3.5 Completeness** (p. 815)
- **14.3.6 Formal Complexity Theory** (p. 821)
- **Exercises** (p. 824)
- **14.4 Chapter Summary** (p. 825)
- **Answers to Selected Exercises** (p. 827)
- **Bibliography** (p. 915)
- **Greek Alphabet** (p. 921)
- **Symbol Glossary** (p. 923)
- **Index** (p. 929)