

Contents

Chapter I The Logic of Compound Statements

I

1.1 Logical Form and Logical Equivalence 1

Identifying logical form; Statements; Logical connectives: not, and, and or; Translation to and from symbolic notation; Statement forms and their truth tables; Exclusive or; Logical equivalence; De Morgan's laws; Tautologies and contradictions; Summary of logical equivalences

1.2 Conditional Statements 17

Conditional statements; The negation of a conditional statement; Contrapositive; Converse and inverse; Only if and the biconditional; Necessary and sufficient conditions

1.3 Valid and Invalid Arguments 28

Arguments, argument forms and validity; Modus ponens and modus tollens; Other argument forms; Converse and inverse errors; Relation of the validity or invalidity of an argument and the truth or falsity of its conclusion; contradiction rule; Application to solving logical puzzles

1.4 Application: Digital Logic Circuits 41

Relation between switching circuits and Boolean expressions; Black boxes and gates; Circuits, input/output tables, and Boolean expressions; Simplifying circuits

1.5 Application: Number Systems and Circuits for Addition 57

Binary representation of numbers; Binary addition and subtraction; Circuits for computer addition; Two's complements and the computer representation of negative integers; Hexadecimal notation

Chapter 2 The Logic of Quantified Statements

75

2.1 Predicates and Quantified Statements I 75

Predicates; Set notation; Universal and existential statements; Translating between formal and informal language; Universal conditional statements; Equivalent forms of universal and existential statements; Implicit

v

quantification; Negations of universal and existential statements; Negations of universal conditional statements; Vacuous truth of universal statements

2.2 Predicates and Quantified Statements II 89

Alternate forms for universal conditional statements; Statements containing both “for all” and “there exists;” Relation among \forall , \exists , \wedge , and \vee ; The use of predicates in Prolog

2.3 Arguments with Quantified Statements 99

Valid argument forms and arguments; Rule of universal instantiation; Universal modus ponens and universal modus tollens; Proving validity; Using diagrams to test for validity; Converse and inverse errors

Chapter 3 Elementary Number Theory and Methods of Proof 112

3.1 Direct Proof and Counterexample I: Introduction 113

Introduction to the basic techniques of direct proof and disproof by counterexample; Properties of even and odd integers and prime and composite numbers

3.2 Direct Proof and Counterexample II: Rational Numbers 125

Exploring the definition and properties of rational numbers

3.3 Direct Proof and Counterexample III: Divisibility 131

Definition of divisibility; Examples and properties; The unique factorization theorem

3.4 Direct Proof and Counterexample IV: Division into Cases and the Quotient-Remainder Theorem 140

Discussion of the quotient-remainder theorem and examples; div and mod; Alternate representations of integers and applications in number theory

3.5 Direct Proof and Counterexample V: Floor and Ceiling 148

Definition and basic properties of the floor and ceiling of a number; The floor of $n/2$

3.6 Indirect Argument: Contradiction and Contraposition 154

Proof by contradiction; There is no greatest integer; The sum of a rational number and an irrational number; Proof by contraposition; When the square of an integer is even

3.7 Two Classical Theorems 161*The irrationality of $\sqrt{2}$; The infinitude of the primes***3.8 Application: Algorithms 166***Notation for algorithms; Trace tables; The division algorithm; The Euclidean algorithm***Chapter 4 Sequences and Mathematical Induction****180****4.1 Sequences 180***Terminology of sequences; Explicit formula for a sequence; Examples; Finding an explicit formula from initial terms; Summation notation; Telescoping sums; Transforming a sum by a change of variable; Product notation; Properties of summations and products; Factorial notation; One-dimensional arrays; Algorithm to change from decimal to binary notation***4.2 Mathematical Induction I 194***Principle of mathematical induction; Sum of the first n integers; Sum of a geometric sequence***4.3 Mathematical Induction II 205***Comparison of mathematical induction and inductive reasoning; Proving divisibility properties; Proving inequalities***4.4 Strong Mathematical Induction and the Well-Ordering Principle 212***Explanation and examples including proof that every integer greater than 1 is divisible by a prime, that a sequence has a certain property, that any parenthesization of a product of n distinct factors results in $n - 1$ multiplications, and that every positive integer has a unique binary representation; The well-ordering principle; Proof of the quotient-remainder theorem***4.5 Application: Correctness of Algorithms 221***Meaning of program correctness; General format; pre-conditions and post-conditions; Loop invariants and the loop invariant theorem; Correctness of a loop to compute a product; Correctness of the division algorithm and the Euclidean algorithm*

Chapter 5 Set Theory 231

5.1 Basic Definitions of Set Theory 231

Definition of subset; Venn diagrams; Relations among sets of numbers; Distinction between \subseteq and \in ; Definitions of equality, union, intersection, difference, and complements of sets; Cartesian products; Formal language; Algorithm for checking a subset relation

5.2 Properties of Sets 244

List of basic set properties; How to prove set properties using element arguments (via procedural versions of definitions); Disproving proposed set properties; Deriving additional set properties from those on a basic list

5.3 The Empty Set, Partitions, Power Sets, and Boolean Algebras 258

How to prove a set is empty; Set properties that involve the empty set; Partitions of sets; Power sets; Boolean algebras

5.4 Russell's Paradox and the Halting Problem 268

The barber puzzle; Russell's paradox; The halting problem

Chapter 6 Counting 273

6.1 Counting and Probability 274

Concept of sample space; Probability in the equally likely outcomes case; Tossing coins, rolling dice; Counting the elements of lists, sublists, and one-dimensional arrays

6.2 Possibility Trees and the Multiplication Rule 281

Possibility trees; The multiplication rule; Counting possibilities with and without repetition; Permutations; Permutations of selected elements: r -permutations; Proving properties of $P(n, r)$.

6.3 Counting Elements of Disjoint Sets: The Addition Rule 295

The addition rule; The difference rule; The inclusion/exclusion rule; The number of elements in a general union, the number of elements in an intersection

6.4 Counting Subsets of a Set: Combinations 306

R -combinations; Ordered and unordered selections; Relation between permutations and combinations; Permutations of a set with repeated elements; A common mistake: double counting

6.5 R-Combinations with Repetition Allowed 322

R-combinations with repetition allowed; Multisets; How to count these; Applications

6.6 The Algebra of Combinations 330

Combinatorial formulas; New formulas from old by substitution; Pascal's triangle; Algebraic and combinatorial proofs of Pascal's formula

6.7 The Binomial Theorem 336

The binomial theorem; Algebraic and combinatorial proofs; Applications

Chapter 7 Functions**344****7.1 Functions Defined on General Sets 344**

Definition of a function as a relationship between the elements of two sets; Arrow diagram of a function; Function machines; Equality of functions; Examples of functions such as the identity function, sequences, functions defined on a power set, functions defined on a language, logarithmic functions, encoding and decoding functions, and Hamming distance function; Boolean functions; Determining whether a function is well-defined

7.2 Application: Finite-State Automata 357

Definitions and examples of finite-state automata; How to construct a finite-state automaton to do a certain job; The language accepted by a finite-state automaton; The eventual-state function; Simulating a finite-state automaton using software

7.3 One-to-One and Onto, Inverse Functions 369

Definition and examples of one-to-one and onto functions; Application to hash functions; Properties of logarithmic and exponential functions; One-to-one correspondences; Inverse functions

7.4 Application: The Pigeonhole Principle 387

Statement and discussion of principle; Applications; Generalized pigeonhole principle and applications; Proof of the pigeonhole principle

7.5 Composition of Functions 401

Definition and examples; Theorems on composition of one-to-one and onto functions

7.6 Cardinality with Applications to Computability 411

Definition of cardinality and countability; Countability of the set of all integers, the set of all even integers, and the set of all rational numbers; Uncountability of the real numbers; Countability of the set of all computer programs in a given computer language; Impossibility of computing certain functions

Chapter 8 Recursion 424

8.1 Recursively Defined Sequences 424

Definition of recurrence relation; Computing terms of recursively defined sequences; The towers of Hanoi; The Fibonacci numbers; Compound interest; Number of bit strings with a certain property; Number of partitions of a set into r subsets; Stirling numbers of the second kind

8.2 Solving Recurrence Relations by Iteration 441

Finding explicit formulas for recursively defined sequences by iteration; Arithmetic and geometric sequences; Using mathematical induction to check whether a recursively defined sequence satisfies a given explicit formula

8.3 Second-Order Linear Homogeneous Recurrence Relations with Constant Coefficients 453

Technique for solving these special relations; Formula for the Fibonacci sequence; Gambler's ruin

8.4 General Recursive Definitions 466

Recursive definition of Boolean expressions, parentheses structures, arithmetic expressions, Σ , Π , and finite unions and intersections; Recursively defined functions

Chapter 9 O-Notation and the Efficiency of Algorithms 476

9.1 Real-Valued Functions of a Real Variable and Their Graphs 476

Graph of a function; Graphs of integral and fractional power functions; Graph of the floor function; Graphs of functions defined on sets of integers; The graph of a multiple of a function; Increasing and decreasing functions

9.2 O-Notation 485

Definition of order; Graphical interpretation; Computing orders of functions from the definition; Orders of polynomial functions; Orders of rational functions; "Best" big-oh approximation

9.3 Application: Efficiency of Algorithms I 495

Use of the order notation to discuss algorithm efficiency; Computing orders of simple algorithms; Calculating the efficiency of the sequential search, insertion sort, and selection sort algorithms; Comparing algorithms for polynomial evaluation

**9.4 Exponential and Logarithmic Functions:
Graphs and Orders 505**

Graphs of logarithmic and exponential functions; Consequences of the fact that the logarithmic function with base $b > 1$ is increasing; The number of bits needed to represent an integer in binary notation; Using logarithms to solve a recurrence relation; Exponential and logarithmic orders

9.5 Application: Efficiency of Algorithms II 519

Divide-and-conquer algorithms; Calculating the efficiency of the binary search and merge sort algorithms

Chapter 10 Relations 533**10.1 Relations on Sets 533**

Definition and notation for relations; Examples of relations; Inverse of a relation; Arrow diagram of a relation; Functions and relations; Directed graph of a relation; n -ary relations and relational databases

10.2 Reflexivity, Symmetry, and Transitivity 546

Reflexive, symmetric, and transitive properties; Transitive closure of a relation; Properties of relations on infinite sets

10.3 Equivalence Relations 555

The relation induced by a partition; Equivalence relations; Examples such as congruence classes modulo n and equivalence of digital logic circuits; Equivalence classes; Lemma that two elements are equivalent if, and only if, they are in the same class; Theorem on the partition of a set by an equivalence relation; Examples of equivalence classes

10.4 Application: Simplifying Finite-State Automata 572

An equivalence relation on the set of states of a finite-state automaton; The quotient automaton; Equivalent automata

10.5 Partial Order Relations 585

Definition and examples; Lexicographic order; Hasse diagrams; Partially and totally ordered sets; Topological sorting; PERT and CPM

Chapter 11 Graphs and Trees 602**11.1 Graphs: An Introduction 602**

Basic terminology and examples of graphs and directed graphs (communication network, representation of a knowledge system, state graph); Special graphs (simple graphs, complete graphs, bipartite graphs); Subgraphs; The concept of degree; Relation of total degree to number of edges; Applications

11.2 Paths and Circuits 619

The puzzle of the Königsberg bridges; Basic definitions of walks, paths, and circuits; Connectedness; Euler circuits; Euler's theorem; Algorithm for constructing an Euler circuit; Hamiltonian circuits; The traveling salesperson problem

11.3 Matrix Representations of Graphs 640

Matrix notation; Adjacency matrices of directed and undirected graphs; Matrices and connected components; Matrix multiplication; Using matrix entries to find the number of walks of length n in a graph

11.4 Isomorphisms of Graphs 656

Definition of graph isomorphism; Examples; Finding all nonisomorphic graphs with certain properties; isomorphic invariants; Using isomorphic invariants to show that graphs are not isomorphic; Graph isomorphism for simple graphs

11.5 Trees 664

Definition and examples of trees (decision tree, derivation tree, structure of hydrocarbon molecules); Equivalent characterizations of trees; Determining number of trees with certain properties; Rooted trees; Binary trees

11.6 Spanning Trees 683

Definition of a spanning tree; Proof of existence; Kruskal's and Prim's algorithms for finding the minimal spanning tree of a weighted graph

Appendix A Properties of the Real Numbers 695**Appendix B Solutions and Hints to Selected Exercises 698**

Photo Credits 800

Index I-1