A Brief Outline of Discrete Mathematics for the Undergraduate Computer Science Student

Dale Fletter

February 21, 2019

Contents

1 The Foundations: Logic and Proofs **1**

1.1 Propositional Logic . . . 2
- 1.1.1 Introduction . . . 2
- 1.1.2 Propositions . . . 2
- 1.1.3 Conditional Statements . . . 2
- 1.1.4 Truth Tables of Compound Propositions . . . 2
- 1.1.5 Precedence of Logical Operations . . . 2
- 1.1.6 Logic and Bit Operations . . . 2

1.2 Applications of Propositional Logic . . . 2
- 1.2.1 Introduction . . . 2
- 1.2.2 Translating English Sentences . . . 2
- 1.2.3 System Specifications . . . 2
- 1.2.4 Boolean Searches . . . 2
- 1.2.5 Logic Puzzles . . . 2
- 1.2.6 Logic Circuits . . . 2

1.3 Propositional Equivalences . . . 2
- 1.3.1 Introduction . . . 2
- 1.3.2 Logical Equivalneces . . . 2
- 1.3.3 Using De Morgan's Laws . . . 2
- 1.3.4 Constructing New Logical Equivalences . . . 2
- 1.3.5 Propositional Satisfiability . . . 2
- 1.3.6 Applications of Satisfiability . . . 2
- 1.3.7 Solving Satisfiability Problems . . . 2

1.4 Predicates and Quantifiers . . . 2
- 1.4.1 Introduction . . . 2
- 1.4.2 Predicates . . . 2
- 1.4.3 Quantifiers . . . 2
- 1.4.4 Quantifiers with Restricted Domains . . . 2
- 1.4.5 Precedence of Quantifiers . . . 2
- 1.4.6 Binding Variables . . . 2

1.4.7 Logical Equivalences Involving Quantifiers . . . 2
1.4.8 Negating Quantified Expressions . . . 2
1.4.9 Translating from English into Logical Expressions . . . 2
1.4.10 Using Quantifiers in System Specifications . . . 2
1.4.11 Examples from Lewis Carroll . . . 2
1.4.12 Logic Programming . . . 2
1.5 Nested Quantifiers . . . 2
1.5.1 Introduction . . . 2
1.5.2 Understanding Statements Involving Nested Quantifiers . . . 2
1.5.3 The Order of Quantifiers . . . 2
1.5.4 Translating Mathematical Statements into Statements Involving Nested Quantifiers . . . 2
1.5.5 Translating from Nested Quantifiers into English . . . 2
1.5.6 Translating English Sentences into Logical Expressions . . . 2
1.5.7 Negating Nested Quantifiers . . . 2
1.6 Rules of Inference . . . 2
1.6.1 Introduction . . . 2
1.6.2 Valid Arguments in Propositional Logic . . . 2
1.6.3 Rules of Inference for Propositional Logic . . . 2
1.6.4 Using Rules of Inference to Build Arguments . . . 2
1.6.5 Resolution . . . 2
1.6.6 Fallacies . . . 2
1.6.7 Rules of Inference for Qualtified Statements . . . 2
1.6.8 Combining Rules of Inference for Propositions and Quantified Statements . . . 2
1.7 Introduction to Proofs . . . 2
1.7.1 Introduction . . . 2
1.7.2 Some Terminology . . . 2
1.7.3 Understanding How Theorems Are Stated . . . 2
1.7.4 Methods of Proving Theorems . . . 2
1.7.5 Direct Proofs . . . 2
1.7.6 Proof by Contraposition . . . 2
1.7.7 Proofs by Contradiction . . . 2
1.7.8 Mistakes in Proofs . . . 2
1.7.9 Just a Beginning . . . 2
1.8 Proof Methods and Strategy . . . 2
1.8.1 Introduction . . . 2
1.8.2 Exhaustive Proof and Proof by Cases . . . 2
1.8.3 Existence Proofs . . . 2
1.8.4 Uniqueness Proofs . . . 2
1.8.5 Proof Strategies . . . 2

1.8.6 Looking for Counterexamples . . . 2
1.8.7 Proof Strategy in Action . . . 2
1.8.8 Tilings . . . 2
1.8.9 The Role of Open Problems . . . 2
1.8.10 Additional Proof Methods . . . 2

2 Basic Structures: Sets, Functions, Sequences, Sums, and Matrices 3
2.1 Sets . . . 4
2.1.1 Introduction . . . 4
2.1.2 Venn Diagrams . . . 4
2.1.3 Subsets . . . 4
2.1.4 The Size of a Set . . . 4
2.1.5 Power Sets . . . 4
2.1.6 Cartesian Products . . . 4
2.1.7 Using Set Notation with Quantifiers . . . 4
2.1.8 Truth Sets and Quantifiers . . . 4
2.2 Set Operations . . . 4
2.2.1 Introduction . . . 4
2.2.2 Set Identities . . . 4
2.2.3 Generalized Unions and Intersections . . . 4
2.2.4 Computer Representation of Sets . . . 4
2.3 Functions . . . 4
2.3.1 Introduction . . . 4
2.3.2 One-to-One and Onto Functions . . . 4
2.3.3 Inverse Functions and Compositions of Functions . . . 4
2.3.4 The Graphs of Functions . . . 4
2.3.5 Partial Functions . . . 4
2.4 Sequences and Summations . . . 4
2.4.1 Introduction . . . 4
2.4.2 Sequences . . . 4
2.4.3 Recurrence Relations . . . 4
2.4.4 Special Integer Sequences . . . 4
2.4.5 Summations . . . 4
2.5 Cardinality of Sets . . . 4
2.5.1 Introduction . . . 4
2.5.2 Countable Sets . . . 4
2.5.3 An Uncountable Set . . . 4
2.6 Matrices . . . 4
2.6.1 Introduction . . . 4
2.6.2 Matrix Arithmetic . . . 4
2.6.3 Transposes and Powers of Matrices . . . 4

2.6.4 Zero-One Matrices 4

3 Algorithms **5**
3.1 Algorithms 6
3.1.1 Introduction 6
3.1.2 Searching Algorithms 6
3.1.3 Sorting 6
3.1.4 Greedy Algorithms 6
3.1.5 The Halting Problem 6
3.2 The Growth of Functions 6
3.2.1 Introduction 6
3.2.2 Big-O Notation 6
3.2.3 Big-O Estimates for Some Important Functions 6
3.2.4 The Growth of Cobinations of Functions 6
3.2.5 Big-Omega and Big-Theta Notation 6
3.3 Complexity of Algorithms 6
3.3.1 Introduction 6
3.3.2 Time Complexity 6
3.3.3 Complexity of Matrix Multiplication 6
3.3.4 Algorithmic Paradigms 6
3.3.5 Understanding the Complexity of Algorithms 6

4 Number Theory and Cryptography **7**
4.1 Divisibility and Modular Arithmetic 8
4.1.1 Introduction 8
4.1.2 Division 8
4.1.3 The Division Algorithm 8
4.1.4 Modular Arithmetic 8
4.1.5 Arithmetic Modulo m 8
4.2 Integer Representation and Algorithms 8
4.2.1 Introduction 8
4.2.2 Representations of Integers 8
4.2.3 Algorithms for Integer Operations 8
4.2.4 Modular Exponentiation 8
4.3 Primes and Greates Common Divisors 8
4.3.1 Introduction 8
4.3.2 Primes 8
4.3.3 Trial Division 8
4.3.4 The Sieve of Eratosthenes 8
4.3.5 Conjectures and Open Problems About Primes 8
4.3.6 Greatest Common Divisors and Least Common Multiples 8

4.3.7 The Euclidean Algorithm . . . 8
4.4 Solving Congruences . . . 8
4.4.1 Introduction . . . 8
4.4.2 Linear Congruences . . . 8
4.4.3 The Chinese Remainder Theorem . . . 8
4.4.4 Computer Arithmetic with Large Integers . . . 8
4.4.5 Fermat's Little Theorem . . . 8
4.4.6 Pseudoprimes . . . 8
4.4.7 Primitive Roots and Discrete Logarithms . . . 8
4.5 Applications of Congruences . . . 8
4.5.1 Hashing Functions . . . 8
4.5.2 Pseudorandom Numbers . . . 8
4.5.3 Check Digits . . . 8
4.6 Cryptography . . . 8
4.6.1 Introduction . . . 8
4.6.2 Classical Cryptography . . . 8
4.6.3 PublicKey Cryptography . . . 8
4.6.4 The RSA Cryptosystem . . . 8
4.6.5 RSA Encryption . . . 8
4.6.6 RSA Dcryption . . . 8
4.6.7 RSA as a Public Key System . . . 8
4.6.8 Cryptographic Protocols . . . 8

5 Induction and Recursion 9
5.1 Mathematical Induction . . . 10
5.1.1 Introduction . . . 10
5.1.2 Mathematical Induction . . . 10
5.1.3 Why Mathematical Induction is Valid . . . 10
5.1.4 The Good and the Bad of Mathematical Induction . . . 10
5.1.5 Examples of Proofs by Mathematical Induction . . . 10
5.1.6 Mistaken Proofs By Mathematical Induction . . . 10
5.1.7 Guidelines for Proofs by Mathematical Induction . . . 10
5.2 Strong Induction and Well-Ordering . . . 10
5.2.1 Introduction . . . 10
5.2.2 Strong Induction . . . 10
5.2.3 Examples of Proofs Using Strong Induction . . . 10
5.2.4 Using Strong Induction in Computation Geometry . . . 10
5.2.5 Proofs Using the Well-Ordered Property . . . 10
5.3 Recursive Definitions and Structural Induction . . . 10
5.3.1 Introduction . . . 10
5.3.2 Recursively Defined Functions . . . 10

5.3.3 Recursively Defined Sets and Structures . . . 10
5.3.4 Structural Induction . . . 10
5.3.5 Generalized Induction . . . 10
5.4 Recursive Algorithms . . . 10
5.4.1 Induction . . . 10
5.4.2 Proving Recursive Algorithms Correct . . . 10
5.4.3 Recursion and Iteration . . . 10
5.4.4 The Merge Sort . . . 10
5.5 Program Correctness . . . 10
5.5.1 Introduction . . . 10
5.5.2 Program Verification . . . 10
5.5.3 Rules of Inference . . . 10
5.5.4 Conditional Statements . . . 10
5.5.5 Loop Invariants . . . 10

6 Counting **11**
6.1 The Basics of Counting . . . 12
6.1.1 Introduction . . . 12
6.1.2 Basic Counting Principles . . . 12
6.1.3 More Complex Counting Problems . . . 12
6.1.4 The Subtraction Rule (Inclusion-Exclusion for Two Sets) . . . 12
6.1.5 The Division Rule . . . 12
6.1.6 Tree Diagrams . . . 12
6.2 The Pigeonhole Principle . . . 12
6.2.1 Introduction . . . 12
6.2.2 The Generalized Pigeonhole Principle . . . 12
6.2.3 Some Elegant Applications of the Pigeonhold Principle . . . 12
6.3 Permutations and Combinations . . . 12
6.3.1 Introduction . . . 12
6.3.2 Permutations . . . 12
6.3.3 Combinations . . . 12
6.4 Binomial Coefficients and Indentities . . . 12
6.4.1 The Binomial Theorem . . . 12
6.4.2 Pascal's Identify and Triangle . . . 12
6.4.3 Other Identities Involving Binomial Coefficients . . . 12
6.5 Generalized Permutations and Combinations . . . 12
6.5.1 Introduction . . . 12
6.5.2 Permutations with Repetition . . . 12
6.5.3 Combinations with Repetition . . . 12
6.5.4 Permutations with Indistinguishable Objects . . . 12
6.5.5 Distribuing Objects into Boxes . . . 12

6.6 Generating Permutations and Combinations 12
6.6.1 Introduction 12
6.6.2 Generating Permutations 12
6.6.3 Generating Combinations 12

7 Discrete Probability **13**
7.1 An Introduction to Discrete Probability 14
7.1.1 Introduction 14
7.1.2 Finite Probability 14
7.1.3 Probabilities of Complements and Unions of Events 14
7.1.4 Probabilistic Reasoning 14
7.2 Probability Theory 14
7.2.1 Introduction 14
7.2.2 Assigning Probabilities 14
7.2.3 Probabilities of Complements and Unions of Events 14
7.2.4 Conditional Probability 14
7.2.5 Independence 14
7.2.6 Bernoulli Trials and the Binomial Distribution 14
7.2.7 Random Variables 14
7.2.8 The Birthday Problem 14
7.2.9 Monte Carlo Algorithms 14
7.2.10 The Probabilistic Method 14
7.3 Bayes' Theorem 14
7.3.1 Introduction 14
7.3.2 Bayes' Theorem 14
7.3.3 Bayesian Spam Filters 14
7.4 Expected Value and Variance 14
7.4.1 Introduction 14
7.4.2 Expected Values 14
7.4.3 Linearity of Expectations 14
7.4.4 Average-Case Computational Complexity 14
7.4.5 The Geometric Distribution 14
7.4.6 Independent Random Variables 14
7.4.7 Variance 14

8 Advanced Counting Techniques **15**
8.1 Application of Recurrence Relations 16
8.1.1 Introduction 16
8.1.2 Modeling With Recurrence Relations 16
8.1.3 Algorithms and Recurrence Relations 16
8.2 Solving Linear Recurrence Relations 16

8.2.1 Introduction . . . 16
8.2.2 Solving Linear Homogeneous Recurrence Relations with Constant Coefficients . . . 16
8.2.3 Linear Nonhomogeneous Recurrence Relations with Constant Coefficients . . . 16
8.3 Divide-and-Conquer Algorithms and Recurrence Relations . . . 16
8.3.1 Introduction . . . 16
8.3.2 Divide-and-Conquer Recurrence Relations . . . 16
8.4 Generating Functions . . . 16
8.4.1 Introduction . . . 16
8.4.2 Useful Facts About Power Series . . . 16
8.4.3 Counting Problems and Generating Functions . . . 16
8.4.4 Using Generating Functions to Solve Recurrence Relations . . . 16
8.4.5 Proving Indentities via Generating Functions . . . 16
8.5 Inclusion-Exclusion . . . 16
8.5.1 Introduction . . . 16
8.5.2 The Principle of Inclusion-Exclusion . . . 16
8.6 Applications of Inclusion-Exclusion . . . 16
8.6.1 Introduction . . . 16
8.6.2 An Alternative Form of Inclusion-Exclusion . . . 16
8.6.3 The Sieve of Eratosthenes . . . 16
8.6.4 The Number of Onto Functions . . . 16
8.6.5 Derangements . . . 16

9 Relations **17**
9.1 Relations and Their Properties . . . 18
9.1.1 Introduction . . . 18
9.1.2 Functions as Relations . . . 18
9.1.3 Relations on a Set . . . 18
9.1.4 Combining Relations . . . 18
9.2 n-ary Relations and Their Applications . . . 18
9.2.1 Introduction . . . 18
9.2.2 n-ary Relations . . . 18
9.2.3 Databases and Relations . . . 18
9.2.4 Operations on n-ary Relations . . . 18
9.2.5 SQL . . . 18
9.3 Representing Relations . . . 18
9.3.1 Introduction . . . 18
9.3.2 Representing Relations Using Matrices . . . 18
9.3.3 Representing Relations Using Digraphs . . . 18
9.4 Closure of Relations . . . 18

9.4.1 Introduction . . . 18
9.4.2 Closures . . . 18
9.4.3 Paths in Directed Graphs . . . 18
9.4.4 Transitive Closures . . . 18
9.4.5 Warshall's Algorithm . . . 18
9.5 Equivalence Relations . . . 18
9.5.1 Introduction . . . 18
9.5.2 Equivalence Relations . . . 18
9.5.3 Equivalence Classes . . . 18
9.5.4 Equivalence Classes and Partitions . . . 18
9.6 Partial Orderings . . . 18
9.6.1 Introduction . . . 18
9.6.2 Lexicographic Order . . . 18
9.6.3 Hasse Diagrams . . . 18
9.6.4 Maximal and Minimal Elements . . . 18
9.6.5 Lattices . . . 18
9.6.6 Topological Sorting . . . 18

10 Graphs **19**
10.1 Graphs and Graph Models . . . 20
10.1.1 Graph Models . . . 20
10.2 Graph Terminology and Special Types of Graphs . . . 20
10.2.1 Introduction . . . 20
10.2.2 Basic Terminology . . . 20
10.2.3 Bipartite Graphs . . . 20
10.2.4 Bipartite Graphs and Matchings . . . 20
10.2.5 Some Applications of Special Types of Graphs . . . 20
10.2.6 New Graphs from Old . . . 20
10.3 Representing Graphs and Graph Isomorphism . . . 20
10.3.1 Introduction . . . 20
10.3.2 Representing Graphs . . . 20
10.3.3 Adjacency Matrices . . . 20
10.3.4 Incidence Matrices . . . 20
10.3.5 Determing whether Two Simple Graphs are Isomorphic . . . 20
10.4 Connectivity . . . 20
10.4.1 Introduction . . . 20
10.4.2 Paths . . . 20
10.4.3 Connectedness in Unidirected Graphs . . . 20
10.4.4 How Connected is a Graph? . . . 20
10.4.5 Connectedness in Directed Graphs . . . 20
10.4.6 Paths and Isomorphism . . . 20

10.4.7 Counting Paths Between Vertices 20
10.5 Euler and Hamilton Paths 20
10.5.1 Introduction 20
10.5.2 Euler Paths and Circuits 20
10.5.3 Hamilton Paths and Circuits 20
10.5.4 Applications of Hamilton Circuits 20
10.6 Shortest-Path Problems 20
10.6.1 Introduction 20
10.6.2 A Shortest-Path Algorithm 20
10.6.3 The Traveling Salesperson Problem 20
10.7 Planar Graphs 20
10.7.1 Introduction 20
10.7.2 Kuratowski's Theorem 20
10.8 Graph Coloring 20
10.8.1 Introduction 20
10.8.2 Applications of Graph Colorings 20

11 Trees **21**
11.1 Introduction to Trees 22
11.1.1 Rooted Trees 22
11.1.2 Trees as Models 22
11.1.3 Properties of Trees 22
11.2 Applications of Trees 22
11.2.1 Introduction 22
11.2.2 Binary Search Trees 22
11.2.3 Decision Trees 22
11.2.4 Game Trees 22
11.3 Tree Traversal 22
11.3.1 Introduction 22
11.3.2 Universal Address Systems 22
11.3.3 Traversal Algorithms 22
11.3.4 Infix, Prefix, and Postfix Notation 22
11.4 Spanning Trees 22
11.4.1 Introduction 22
11.4.2 Depth-First Search 22
11.4.3 Breadth-First Search 22
11.4.4 Backtracking Applications 22
11.4.5 Depth-First Search in Directed Graphs 22
11.5 Minimum Spanning Trees 22
11.5.1 Introduction 22
11.5.2 Algorithms for Minimum Spanning Trees 22

12 Boolean Algebra **23**
12.1 Boolean Functions . . . 24
12.1.1 Introduction . . . 24
12.1.2 Boolean Expressions and Boolean Functions . . . 24
12.1.3 Identities of Boolean Algebra . . . 24
12.1.4 Duality . . . 24
12.1.5 The Abstract Definition of a Boolean Algebra . . . 24
12.2 Representing Boolean Functions . . . 24
12.2.1 Sum-of-Products Expansions . . . 24
12.2.2 Functional Completeness . . . 24
12.3 Logic Gates . . . 24
12.3.1 Introduction . . . 24
12.3.2 Combinations of Gates . . . 24
12.3.3 Examples of Circuits . . . 24
12.3.4 Adders . . . 24
12.4 Minimization of Circuits . . . 24
12.4.1 Introduction . . . 24
12.4.2 Karnaugh Maps . . . 24
12.4.3 Don't Care Conditions . . . 24
12.4.4 The Quine-McCluskey Method . . . 24

13 Modeling Computation **25**
13.1 Languages and Grammars . . . 26
13.1.1 Introduction . . . 26
13.1.2 Phrase-Structure Grammars . . . 26
13.1.3 Types of Phrase-Structure Grammars . . . 26
13.1.4 Derivation Trees . . . 26
13.1.5 Backus-Naur Form . . . 26
13.2 Finite-State Machines with Output . . . 26
13.2.1 Introduction . . . 26
13.2.2 Finite-State Machines with Outputs . . . 26
13.3 Finite-State Machines with No Output . . . 26
13.3.1 Introduction . . . 26
13.3.2 Set of Strings . . . 26
13.3.3 Finite-State Automata . . . 26
13.3.4 Language Recognition by Finite-State Machines . . . 26
13.3.5 Nondeterministic Finite-State Automata . . . 26
13.4 Language Recognition . . . 26
13.4.1 Introduction . . . 26
13.4.2 Kleene's Theorem . . . 26
13.4.3 Regular Sets and Regular Grammars . . . 26

13.4.4 More Powerful Types of Machines 26
13.5 Turing Machines 26
13.5.1 Introduction 26
13.5.2 Definition of Turing Machines 26
13.5.3 Using Turing Machines to Recognize Sets 26
13.5.4 Computing Functions with Turing Machines 26
13.5.5 Different Types of Turing Machines 26
13.5.6 The Church-Turing Thesis 26
13.5.7 Computational Complexity, Computability, and Decidability 26

Chapter 1

The Foundations: Logic and Proofs

1.1 Propositional Logic

1.1.1 Introduction

1.1.2 Propositions

1.1.3 Conditional Statements

1.1.4 Truth Tables of Compound Propositions

1.1.5 Precedence of Logical Operations

1.1.6 Logic and Bit Operations

1.2 Applications of Propositional Logic

1.2.1 Introduction

1.2.2 Translating English Sentences

1.2.3 System Specifications

1.2.4 Boolean Searches

1.2.5 Logic Puzzles

1.2.6 Logic Circuits

1.3 Propositional Equivalences

1.3.1 Introduction

1.3.2 Logical Equivalneces

1.3.3 Using De Morgan's Laws

1.3.4 Constructing New Logical Equivalences

1.3.5 Propositional Satisfiability

1.3.6 Applications of Satisfiability

1.3.7 Solving Satisfiability Problems

1.4 Predicates and Quantifiers

Chapter 2

Basic Structures: Sets, Functions, Sequences, Sums, and Matrices

2.1 Sets

2.1.1 Introduction

Definition of Set

Notation for Common Sets

Definition of Set Equality

The Empty Set

2.1.2 Venn Diagrams

2.1.3 Subsets

Definition of Subset

2.1.4 The Size of a Set

Definition of Set Cardinality

2.1.5 Power Sets

2.1.6 Cartesian Products

2.1.7 Using Set Notation with Quantifiers

2.1.8 Truth Sets and Quantifiers

2.2 Set Operations

2.2.1 Introduction

2.2.2 Set Identities

2.2.3 Generalized Unions and Intersections

2.2.4 Computer Representation of Sets

2.3 Functions

2.3.1 Introduction

Chapter 3

Algorithms

3.1 Algorithms

3.1.1 Introduction

3.1.2 Searching Algorithms

3.1.3 Sorting

3.1.4 Greedy Algorithms

3.1.5 The Halting Problem

3.2 The Growth of Functions

3.2.1 Introduction

3.2.2 Big-O Notation

3.2.3 Big-O Estimates for Some Important Functions

3.2.4 The Growth of Cobinations of Functions

3.2.5 Big-Omega and Big-Theta Notation

3.3 Complexity of Algorithms

3.3.1 Introduction

3.3.2 Time Complexity

3.3.3 Complexity of Matrix Multiplication

3.3.4 Algorithmic Paradigms

3.3.5 Understanding the Complexity of Algorithms

Chapter 4

Number Theory and Cryptography

4.1 Divisibility and Modular Arithmetic

4.1.1 Introduction

4.1.2 Division

4.1.3 The Division Algorithm

4.1.4 Modular Arithmetic

4.1.5 Arithmetic Modulo m

4.2 Integer Representation and Algorithms

4.2.1 Introduction

4.2.2 Representations of Integers

4.2.3 Algorithms for Integer Operations

4.2.4 Modular Exponentiation

4.3 Primes and Greates Common Divisors

4.3.1 Introduction

4.3.2 Primes

4.3.3 Trial Division

4.3.4 The Sieve of Eratosthenes

4.3.5 Conjectures and Open Problems About Primes

4.3.6 Greatest Common Divisors and Least Common Multiples

4.3.7 The Euclidean Algorithm

4.4 Solving Congruences

4.4.1 Introduction

4.4.2 Linear Congruences

4.4.3 The Chinese Remainder Theorem

Chapter 5

Induction and Recursion

5.1 Mathematical Induction

5.1.1 Introduction

5.1.2 Mathematical Induction

5.1.3 Why Mathematical Induction is Valid

5.1.4 The Good and the Bad of Mathematical Induction

5.1.5 Examples of Proofs by Mathematical Induction

5.1.6 Mistaken Proofs By Mathematical Induction

5.1.7 Guidelines for Proofs by Mathematical Induction

5.2 Strong Induction and Well-Ordering

5.2.1 Introduction

5.2.2 Strong Induction

5.2.3 Examples of Proofs Using Strong Induction

5.2.4 Using Strong Induction in Computation Geometry

5.2.5 Proofs Using the Well-Ordered Property

5.3 Recursive Definitions and Structural Induction

5.3.1 Introduction

5.3.2 Recursively Defined Functions

5.3.3 Recursively Defined Sets and Structures

5.3.4 Structural Induction

5.3.5 Generalized Induction

5.4 Recursive Algorithms

5.4.1 Induction

5.4.2 Proving Recursive Algorithms Correct

Chapter 6

Counting

6.1 The Basics of Counting

6.1.1 Introduction

6.1.2 Basic Counting Principles

6.1.3 More Complex Counting Problems

6.1.4 The Subtraction Rule (Inclusion-Exclusion for Two Sets)

6.1.5 The Division Rule

6.1.6 Tree Diagrams

6.2 The Pigeonhole Principle

6.2.1 Introduction

6.2.2 The Generalized Pigeonhole Principle

6.2.3 Some Elegant Applications of the Pigeonhold Principle

6.3 Permutations and Combinations

6.3.1 Introduction

6.3.2 Permutations

6.3.3 Combinations

6.4 Binomial Coefficients and Indentities

6.4.1 The Binomial Theorem

6.4.2 Pascal's Identify and Triangle

6.4.3 Other Identities Involving Binomial Coefficients

6.5 Generalized Permutations and Combinations

6.5.1 Introduction

6.5.2 Permutations with Repetition

6.5.3 Combinations with Repetition

Chapter 7

Discrete Probability

7.1 An Introduction to Discrete Probability

7.1.1 Introduction

7.1.2 Finite Probability

7.1.3 Probabilities of Complements and Unions of Events

7.1.4 Probabilistic Reasoning

7.2 Probability Theory

7.2.1 Introduction

7.2.2 Assigning Probabilities

7.2.3 Probabilities of Complements and Unions of Events

7.2.4 Conditional Probability

7.2.5 Independence

7.2.6 Bernoulli Trials and the Binomial Distribution

7.2.7 Random Variables

7.2.8 The Birthday Problem

7.2.9 Monte Carlo Algorithms

7.2.10 The Probabilistic Method

7.3 Bayes' Theorem

7.3.1 Introduction

7.3.2 Bayes' Theorem

7.3.3 Bayesian Spam Filters

7.4 Expected Value and Variance

7.4.1 Introduction

7.4.2 Expected Values

Chapter 8

Advanced Counting Techniques

8.1 Application of Recurrence Relations

8.1.1 Introduction

8.1.2 Modeling With Recurrence Relations

8.1.3 Algorithms and Recurrence Relations

8.2 Solving Linear Recurrence Relations

8.2.1 Introduction

8.2.2 Solving Linear Homogeneous Recurrence Relations with Constant Coefficients

8.2.3 Linear Nonhomogeneous Recurrence Relations with Constant Coefficients

8.3 Divide-and-Conquer Algorithms and Recurrence Relations

8.3.1 Introduction

8.3.2 Divide-and-Conquer Recurrence Relations

8.4 Generating Functions

8.4.1 Introduction

8.4.2 Useful Facts About Power Series

8.4.3 Counting Problems and Generating Functions

8.4.4 Using Generating Functions to Solve Recurrence Relations

8.4.5 Proving Indentities via Generating Functions

8.5 Inclusion-Exclusion

8.5.1 Introduction

8.5.2 The Principle of Inclusion-Exclusion

8.6 Applications of Inclusion-Exclusion

Chapter 9

Relations

9.1 Relations and Their Properties

9.1.1 Introduction

9.1.2 Functions as Relations

9.1.3 Relations on a Set

9.1.4 Combining Relations

9.2 n-ary Relations and Their Applications

9.2.1 Introduction

9.2.2 n-ary Relations

9.2.3 Databases and Relations

9.2.4 Operations on n-ary Relations

9.2.5 SQL

9.3 Representing Relations

9.3.1 Introduction

9.3.2 Representing Relations Using Matrices

9.3.3 Representing Relations Using Digraphs

9.4 Closure of Relations

9.4.1 Introduction

9.4.2 Closures

9.4.3 Paths in Directed Graphs

9.4.4 Transitive Closures

9.4.5 Warshall's Algorithm

9.5 Equivalence Relations

9.5.1 Introduction

Chapter 10

Graphs

10.1 Graphs and Graph Models

10.1.1 Graph Models

10.2 Graph Terminology and Special Types of Graphs

10.2.1 Introduction

10.2.2 Basic Terminology

10.2.3 Bipartite Graphs

10.2.4 Bipartite Graphs and Matchings

10.2.5 Some Applications of Special Types of Graphs

10.2.6 New Graphs from Old

10.3 Representing Graphs and Graph Isomorphism

10.3.1 Introduction

10.3.2 Representing Graphs

10.3.3 Adjacency Matrices

10.3.4 Incidence Matrices

10.3.5 Determing whether Two Simple Graphs are Isomorphic

10.4 Connectivity

10.4.1 Introduction

10.4.2 Paths

10.4.3 Connectedness in Unidirected Graphs

10.4.4 How Connected is a Graph?

10.4.5 Connectedness in Directed Graphs

10.4.6 Paths and Isomorphism

10.4.7 Counting Paths Between Vertices

Chapter 11

Trees

11.1 Introduction to Trees

11.1.1 Rooted Trees

11.1.2 Trees as Models

11.1.3 Properties of Trees

11.2 Applications of Trees

11.2.1 Introduction

11.2.2 Binary Search Trees

11.2.3 Decision Trees

11.2.4 Game Trees

11.3 Tree Traversal

11.3.1 Introduction

11.3.2 Universal Address Systems

11.3.3 Traversal Algorithms

11.3.4 Infix, Prefix, and Postfix Notation

11.4 Spanning Trees

11.4.1 Introduction

11.4.2 Depth-First Search

11.4.3 Breadth-First Search

11.4.4 Backtracking Applications

11.4.5 Depth-First Search in Directed Graphs

11.5 Minimum Spanning Trees

11.5.1 Introduction

11.5.2 Algorithms for Minimum Spanning Trees

Chapter 12

Boolean Algebra

12.1 Boolean Functions

12.1.1 Introduction

12.1.2 Boolean Expressions and Boolean Functions

12.1.3 Identities of Boolean Algebra

12.1.4 Duality

12.1.5 The Abstract Definition of a Boolean Algebra

12.2 Representing Boolean Functions

12.2.1 Sum-of-Products Expansions

12.2.2 Functional Completeness

12.3 Logic Gates

12.3.1 Introduction

12.3.2 Combinations of Gates

12.3.3 Examples of Circuits

12.3.4 Adders

12.4 Minimization of Circuits

12.4.1 Introduction

12.4.2 Karnaugh Maps

12.4.3 Don't Care Conditions

12.4.4 The Quine-McCluskey Method

Chapter 13

Modeling Computation

13.1 Languages and Grammars

13.1.1 Introduction

13.1.2 Phrase-Structure Grammars

13.1.3 Types of Phrase-Structure Grammars

13.1.4 Derivation Trees

13.1.5 Backus-Naur Form

13.2 Finite-State Machines with Output

13.2.1 Introduction

13.2.2 Finite-State Machines with Outputs

13.3 Finite-State Machines with No Output

13.3.1 Introduction

13.3.2 Set of Strings

13.3.3 Finite-State Automata

13.3.4 Language Recognition by Finite-State Machines

13.3.5 Nondeterministic Finite-State Automata

13.4 Language Recognition

13.4.1 Introduction

13.4.2 Kleene's Theorem

13.4.3 Regular Sets and Regular Grammars

13.4.4 More Powerful Types of Machines

13.5 Turing Machines

13.5.1 Introduction

13.5.2 Definition of Turing Machines