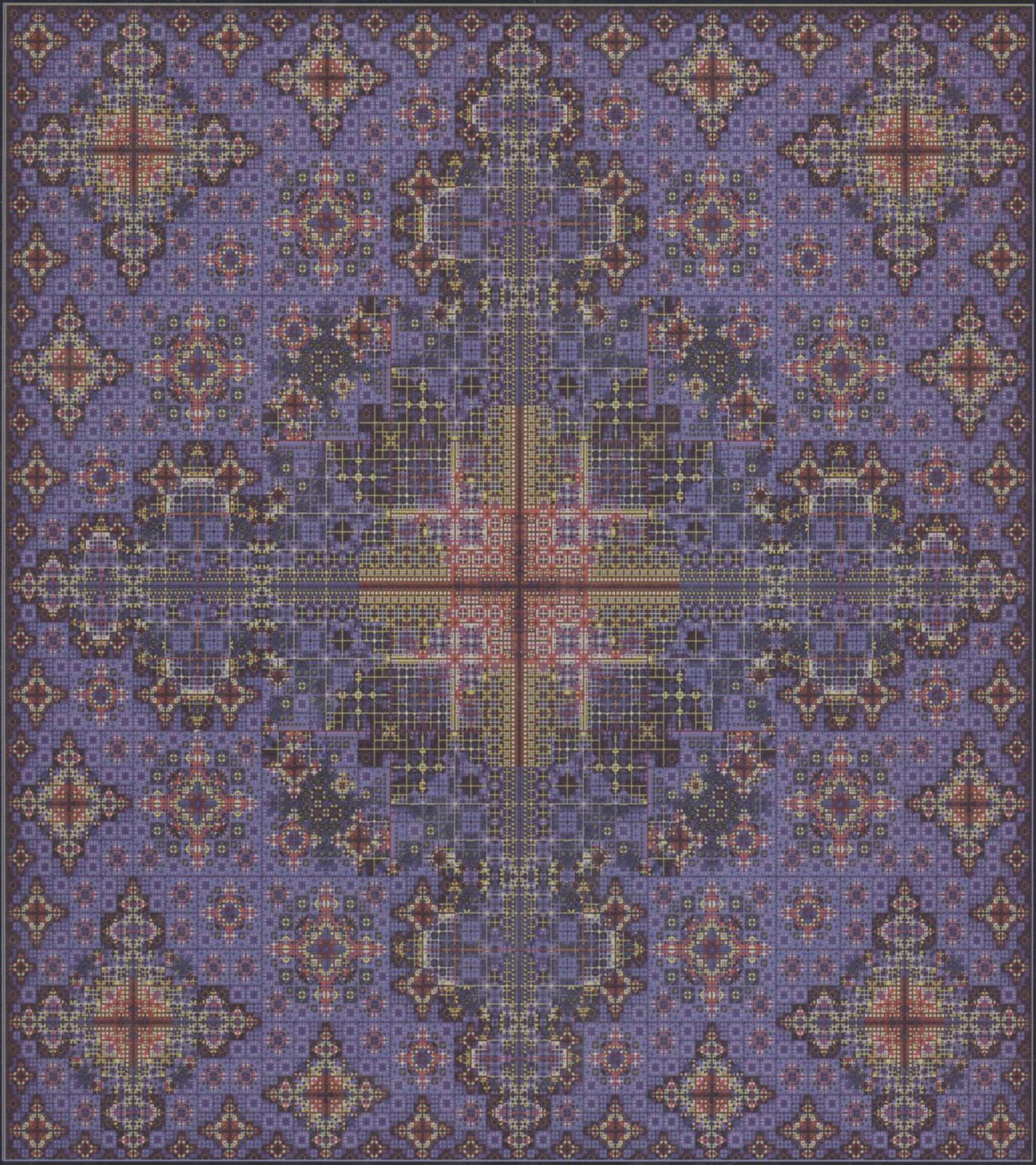
Discrete Mathematics with Proof



Eric Gossett

Discrete Mathematics with Proof

Eric Gossett
Bethel College

Prentice Hall
PEARSON EDUCATION, INC.
Upper Saddle River, New Jersey 07458

Library of Congress Cataloging in Publication Data

Gossett, Eric
Discrete mathematics with proof / Eric Gossett.
p. cm.
Includes bibliographical references and index.
ISBN 0-13-066948-2
1. Mathematics. 2. Computer science–Mathematics. I. Title
QH39.3 .G68 2003 2003
510—dc21 2002032965
CIP

To my wife, Florence Kuofang Gossett,
my daughter, Rachel Shinching Gossett,
and my son, Nathan Mui Gossett.

Editor in Chief: Sally Yagan
Acquisition Editor: George Lobell
Vice President/Director of Production and Manufacturing: David W. Riccardi
Executive Managing Editor: Kathleen Schiaparelli
Senior Managing Editor: Linda Mihatov Behrens
Assistant Managing Editor: Bayani Mendoza de Leon
Production Editor: Jeanne Audino
Manufacturing Buyer: Michael Bell
Manufacturing Manager: Trudy Pisciotti
Marketing Manager: Halee Dinsey
Marketing Assistant: Rachel Beckman
Editorial Assistant/Supplements Editor: Jennifer Brady
Art Director: Heather Scott
Interior Designer: Judith Matz-Coniglio
Cover Designer: Heather Scott
Art Editor: Thomas Benfatti
Assistant Managing Editor, Math Media Production: John Matthews
Creative Director: Carole Anson
Director of Creative Services: Paul Belfanti
Cover Photo: Eric Gossett
Art Studio: Laserwords
Compositor: Dennis Kletzing

Prentice Hall

Pearson Education, Inc.
Upper Saddle River, New Jersey 07458

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-066948-2

Pearson Education LTD.
Pearson Education Australia PTY, Limited
Pearson Education Singapore, Pte. Ltd
Pearson Education North Asia Ltd, Hong Kong
Pearson Education Canada, Ltd., Toronto
Pearson Educacion de Mexico, S.A. de C.V.
Pearson Education–Japan, Inc.
Pearson Education Malaysia, Pte. Ltd

Fundamental Set Properties

Idempotence
$A \cup A = A$
$A \cap A = A$

Associativity
$(A \cup B) \cup C = A \cup (B \cup C)$
$(A \cap B) \cap C = A \cap (B \cap C)$

Commutativity
$A \cup B = B \cup A$
$A \cap B = B \cap A$

Distributivity ($\cap$ **over** $\cup$)
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
$(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$

Complement
$A \cup \overline{A} = U$
$A \cap \overline{A} = \emptyset$

Involution
$\overline{\overline{A}} = A$

Domination
$A \cup U = U$
$A \cap \emptyset = \emptyset$

Identity
$A \cup \emptyset = A$
$A \cap U = A$

De Morgan's Laws
$\overline{A \cup B} = \overline{A} \cap \overline{B}$
$\overline{A \cap B} = \overline{A} \cup \overline{B}$

Distributivity ($\cup$ **over** $\cap$)
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
$(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$

Complement (continued)
$\overline{\emptyset} = U$
$\overline{U} = \emptyset$

Boolean Algebra Axioms

Identity There exist distinct elements, 0 and 1, in B such that for every $x \in B$

$$x + 0 = x$$
$$x \cdot 1 = x.$$

Complement For every $x \in B$, there exists a unique element $\overline{x} \in B$ such that

$$x + \overline{x} = 1$$
$$x \cdot x = 0.$$

Commutativity For every pair of (not necessarily distinct) elements $x, y \in B$

$$x + y = y + x$$
$$x \cdot y = y \cdot x.$$

Distributivity For every three elements $x, y, z \in B$ (not necessarily distinct)

$$x \cdot (y + z) = x \cdot y + x \cdot z$$
$$x + y \cdot z = (x + y) \cdot (x + z).$$

Fundamental Boolean Alegebra Properties

Idempotence $x + x = x$ $x \cdot x = x$	**Domination** $x + 1 = 1$ $x \cdot 0 = 0$
Associativity $(x + y) + z = x + (y + z)$ $(x \cdot y) \cdot z = x \cdot (y \cdot z)$	**De Morgan's Laws** $\overline{x + y} = \overline{x} \cdot \overline{y}$ $\overline{x \cdot y} = \overline{x} + \overline{y}$
Involution $\overline{\overline{x}} = x$	**Absorption** $x + x \cdot y = x$ $x \cdot (x + y) = x$

General Proof Strategies

If the assertion…	Then try…
claims something is true for all integers $n \geq n_0$	mathematical induction
is stated explicitly or implicitly as an implication	direct; indirect; contradiction
contains an existential quantifier	a constructive proof; a non-constructive proof
contains a universal quantifier	finding a counter-example; the choose method
contains the phrase "if and only if"	to prove the two implications separately; to produce a sequence of equivalent statements linking the two sides of the biconditional
is stated as an equivalence	to look for a complete set of implications that are relatively easy to prove
can be easily split into a collection of independent assertions	proof by cases
is an implication with a true conclusion	vacuous proof
is an implication with a false hypothesis	trivial proof
is about membership in a set	direct proof: verify that the element satisfies the set membership requirements
asserts one set is a subset of another	to show that a generic element of the first set is also a member of the second set
asserts the equality of two sets	to show that each set is a subset of the other; to use a sequence of reversible statements with the fundamental set properties and other theorems

Fundamental Logical Equivalences

Idempotence
$(P \vee P) \Leftrightarrow P$
$(P \wedge P) \Leftrightarrow P$

Associativity
$[(P \vee Q) \vee R] \Leftrightarrow [P \vee (Q \vee R)]$
$[(P \wedge Q) \wedge R] \Leftrightarrow [P \wedge (Q \wedge R)]$

Commutativity
$(P \vee Q) \Leftrightarrow (Q \vee P)$
$(P \wedge Q) \Leftrightarrow (Q \wedge P)$

Distributivity ($\wedge$ **over** $\vee$)
$[P \wedge (Q \vee R)] \Leftrightarrow [(P \wedge Q) \vee (P \wedge R)]$
$[(P \vee Q) \wedge R] \Leftrightarrow [(P \wedge R) \vee (Q \wedge R)]$

Law of the Excluded Middle
$[P \vee (\neg P)] \Leftrightarrow \mathbf{T}$

Law of Double Negation (Involution)
$\neg(\neg P) \Leftrightarrow P$

Law of Simplification
$[(P \wedge Q) \rightarrow P] \Leftrightarrow \mathbf{T}$
$[(P \wedge Q) \rightarrow Q] \Leftrightarrow \mathbf{T}$

Domination
$(P \vee \mathbf{T}) \Leftrightarrow \mathbf{T}$
$(P \wedge \mathbf{F}) \Leftrightarrow \mathbf{F}$

Identity
$(P \vee \mathbf{F}) \Leftrightarrow P$
$(P \wedge \mathbf{T}) \Leftrightarrow P$

De Morgan's Laws
$[\neg(P \vee Q)] \Leftrightarrow [(\neg P) \wedge (\neg Q)]$
$[\neg(P \wedge Q)] \Leftrightarrow [(\neg P) \vee (\neg Q)]$

Distributivity ($\vee$ **over** $\wedge$)
$[P \vee (Q \wedge R)] \Leftrightarrow [(P \vee Q) \wedge (P \vee R)]$
$[(P \wedge Q) \vee R] \Leftrightarrow [(P \vee R) \wedge (Q \vee R)]$

Law of Contradiction
$[P \wedge (\neg P)] \Leftrightarrow \mathbf{F}$

Law of Addition
$[P \rightarrow (P \vee Q)] \Leftrightarrow \mathbf{T}$

Logical Equivalences and Rules of Inference for Implication and the Biconditional

Implication
$(P \rightarrow Q) \Leftrightarrow [\neg(P \wedge (\neg Q))] \Leftrightarrow [(\neg P) \vee Q]$

The Biconditional
$(P \leftrightarrow Q) \Leftrightarrow [(P \rightarrow Q) \wedge (Q \rightarrow P)]$
$(P \leftrightarrow Q) \Rightarrow (P \rightarrow Q)$
$(P \leftrightarrow Q) \Rightarrow (Q \rightarrow P)$

Negation Of An Implication
$[\neg(P \rightarrow Q)] \Leftrightarrow [P \wedge (\neg Q)]$

Transitivity of Biconditional
$[(P \leftrightarrow S_1) \wedge (S_1 \leftrightarrow S_2) \wedge \cdots \wedge (S_n \leftrightarrow Q)]$
$\Leftrightarrow (P \leftrightarrow Q)$

Logical Equivalences and Rules of Inference Involving Quantifiers

Negating an Existential Quantifier
$\neg[\exists x \in U, P(x)] \Leftrightarrow [\forall x \in U, \neg P(x)]$

Simplifying Universal to Existential
$[\forall x \in U, P(x) \wedge (U \neq \emptyset)] \Rightarrow [\exists x \in U, P(x)]$

Negating a Universal Quantifier
$\neg[\forall x \in U, P(x)] \Leftrightarrow [\exists x \in U, \neg P(x)]$

Arranging r elements from a set containing n distinct elements

	With Order	Without Order
Without Repetition	$P(n,r) = \frac{n!}{(n-r)!}$	$C(n,r) = \frac{n!}{r! \cdot (n-r)!}$
With Repetition	n^r	$C(n+r-1,r) = \frac{(n+r-1)!}{r! \cdot (n-1)!}$

The number of ways to place n objects into k containers

Objects		Containers: Distinguishable	Containers: Indistinguishable
Distinguishable	$\emptyset$:	k^n	$\sum_{i=1}^{k} S(n,i)$
	$\neg\emptyset$:	$k!S(n,k)$	$S(n,k)$
Indistinguishable	$\emptyset$:	$\binom{k+n-1}{n}$	$\sum_{i=1}^{k} p(n,i)$
	$\neg\emptyset$:	$\binom{n-1}{k-1}$	$p(n,k)$

$\emptyset$: containers may be empty

$\neg\emptyset$: containers must contain at least one object

Some Useful Generating Functions

$G(z)$	Summation Notation	Expanded Notation
$\frac{1}{1-z}$	$\sum_{k=0}^{\infty} z^k$	$1 + z + z^2 + z^3 + \cdots$
$\frac{1}{1+z}$	$\sum_{k=0}^{\infty} (-1)^k z^k$	$1 - z + z^2 - z^3 + \cdots$
$\frac{1}{1-z^m}$	$\sum_{k=0}^{\infty} z^{mk}$	$1 + z^m + z^{2m} + z^{3m} + \cdots$
$\frac{1}{1-cz}$	$\sum_{k=0}^{\infty} c^k z^k$	$1 + cz + c^2z^2 + c^3z^3 + \cdots$
$\frac{1}{(1-z)^m}$	$\sum_{k=0}^{\infty} \binom{m+k-1}{k} z^k$	$1 + mz + \binom{m+1}{2}z^2 + \binom{m+2}{3}z^3 + \cdots$
$\frac{z}{(1-z)^2}$	$\sum_{k=0}^{\infty} k z^k$	$0 + z + 2z^2 + 3z^3 + \cdots$
$(1+z)^c$	$\sum_{k=0}^{\infty} \binom{c}{k} z^k$	$1 + cz + \binom{c}{2}z^2 + \binom{c}{3}z^3 + \cdots$
e^z	$\sum_{k=0}^{\infty} \frac{1}{k!} z^k$	$1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \cdots$

Discrete Mathematics with Proof

Preface

This book has been written for a sophomore-level course in Discrete Mathematics. The material has been directed towards the needs of mathematics and computer science majors, although there is certainly material that is of use for other majors. Students are assumed to have completed a semester of college-level calculus. This assumption is primarily about the level of mathematical maturity of the readers. The material in a calculus course will not often be used in the text.

This textbook has been designed to be suitable for a course that requires students to read the textbook. Many students find this challenging, preferring to just let the instructor tell them "everything they need to know" and using the textbook as a repository of homework exercises and corresponding examples. A typical course in Discrete Mathematics will require much more from the students. Consequently, the textbook needs to support this transition towards greater mathematical maturity.

I have successfully used this text by requiring students to read a section and submit some simple exercises from that section at the start of a class period where I discuss the material for the first time. The following class period, the students will submit more difficult exercises. Consequently, extra care has been taken to ensure that students can follow the presentation in the book even before the material is presented in class. While most instructors do not structure their course in this manner, a textbook that has been written to stand on its own will certainly be of value to the students.

I imagine that this book will work well with a distance education format. However, I feel that personal interaction between the student and the instructor (or a knowledgeable teaching assistant) greatly enhances the learning experience.

DISTINGUISHING CHARACTERISTICS OF THIS TEXT

There are currently many textbooks on the market for a course in Discrete Mathematics. Although there is an assumed common core of topics and level, there is still sufficient variation to provide instructors with viable options for choosing a textbook. Here are some of the features that characterize this book.

- There is a heavy emphasis on proof throughout the text (as indicated by the book's title). The formal setting is introduced in Chapter 2 as sets, logic, and boolean algebras are discussed. Chapter 3 then discusses axiomatic mathematics as a system and subsequently focuses on proof techniques. The proof techniques are extensively illustrated in the rest of the text. For example: proof by contradiction in Chapter 4 with The Halting Problem; constructive proofs in Chapter 8 with "a finite projective plane of order n iff $n - 1$ mutually orthogonal Latin squares of order n"; complete induction in Chapter 3 with the "optimality (for suitors) of the Deferred Acceptance Algorithm". Combinatorial proof is introduced in Chapter 5 and used in Chapter 8 to establish the necessary conditions for the existence of a balanced incomplete block design.

  Many of the more difficult proofs are accompanied by illustrative examples that can be read in parallel with the proof. For instance, Theorem 8.20 on page 486 and Examples 8.49 and 8.50 that appear after the proof.
- The text has been written for students to read actively. The text contains more detailed explanations than some competing texts. Homework problems have been

designed to reinforce reading. Most cannot be completed by merely finding a clone example to copy and modify. The chapters include Quick Check problems at critical points in the reading. These are problems that should be solved before continuing to read. Detailed solutions are presented at the end of the chapter.

- Technology is introduced when it will enhance understanding. For example: a simple perl script for testing regular expressions in Chapter 9; a Java Application (and Applet) that allows students to rubber-band graphs to check for planarity in Chapter 10; several applications that explore the inner workings of recursion in Chapter 7; a Java Application (and Applet) for checking Quine-McCluskey minimizations of binary expressions in Chapter 12. There are also web links to Applets that animate critical algorithms and structures: Boyer-Moore in Chapter 4; finite automata in Chapter 9; logic circuits in Chapter 12. These can all be found at:

  http://www.prenhall.com/gossett/
  or
  http://www.mathcs.bethel.edu/~gossett/DiscreteMathWithProof/

- Combinatorics receives a full chapter (Chapter 8) beyond the standard "combinations and permutations" material presented in Chapter 5. The non-standard topics include Latin squares, finite projective planes, balanced incomplete block designs, coding theory, knapsack problems, Ramsey numbers, partitions, occupancy problems, Stirling numbers, and systems of distinct representatives. The chapter begins with an overview of the major themes that unify the field of combinatorics.
- There are several major examples that present significant algorithms. Examples include: Chapter 1: the Deferred Acceptance Algorithm (the Stable Marriage Problem); Chapter 4: Boyer-Moore algorithm for pattern matching; Chapter 7: recursive algorithms for Sierpinski curves, persian rugs, and adaptive quadrature.

  Other examples cover problems with a significant history. Examples include: Chapter 7: the Josephus problem; Chapter 8: Kirkman's Schoolgirl Problem; Chapter 10: the 5 regular polyhedra and a proof of the Five Color Theorem.

  There are also some important examples from the field of computer science. These include: Chapter 4: The Halting problem; Chapter 9: Shannon's mathematical model of information; Chapter 11: XML; and Chapter 12: Normal Forms in relational databases.
- The Discrete Mathematics course at Bethel College is equally populated with mathematics majors and computer science majors. Consequently, this text was designed to be appropriate for courses for mathematics majors, courses for computer science majors, and courses with bimodal populations. There is sufficient material to design a one-semester course for any of these three options. It is also possible to design a two-semester course that covers the entire book.

  An example of using the text with a bimodal group can be found in the chapter on recursion. Chapter 7 starts with an algorithmic approach and then presents recurrence relations (a mathematical approach). Both sets of students see the concept (recursion) in a form that is oriented towards their own major. In addition, they are exposed to an alternative viewpoint.
- Several other topics receive more coverage than is typical. These include: Chapter 4: expressing algorithms, the Halting Problem; Chapter 6: Bayes' Theorem; several topics in Chapter 8; Chapter 9: regular expressions.
- The text begins with an introductory chapter that provides some explanation and examples of what discrete mathematics is about. This is rare among discrete mathematics textbooks.
- Limitations are discussed where appropriate. For example, the solution of linear homogeneous recurrence relations with constant coefficients that is presented in

Chapter 7 requires the factorization of polynomials. There is no general factorization technique for polynomials of degree 6 or higher. The text also contains guidelines to determine whether recursion is an appropriate solution technique for a particular problem.

- An instructor's solution manual with detailed solutions to every problem is available for use by student graders and professors. Appendix G is a free student's solution manual containing detailed solutions to selected exercises.

TEXT ORGANIZATION

The chapters in the book are briefly summarized in the following paragraphs.

Chapter 1: Introduction

Chapter 1 provides a working definition of *discrete mathematics* and then offers the reader some brief glimpses at some of the topics that will be covered in the remaining chapters. The chapter also introduces the stable marriage problem and the deferred acceptance algorithm. This material is covered in some detail and appears again in several other chapters.

The exposition of the stable marriage problem introduces a non-trivial algorithm and some proofs. The problem, the algorithm, and the proofs are all fairly intuitive. They prepare the reader for the more detailed expositions of algorithms and proofs that will follow in future chapters. The problem also shows the reader that the material in this course may be different from what they have studied in previous mathematics courses.

Chapter 2: Sets, Logic, and Boolean Algebras

Much of the material in this chapter is not what students tend to rate as most interesting. However, it is foundational to much of what follows. It is even more important than in previous decades because many students are now graduating from high school without ever learning the basics of set theory. Many have never been exposed to either the basic terminology (element, union, intersection) or the standard notation ($\in$, $\cup$, $\cap$).

The basic concepts of propositional and predicate logic are introduced in this chapter. They also serve as a basis for the proof strategies introduced in chapter 3.

The basic properties of sets and logic are presented in a similar style to emphasize the similarities. This parallel exposition provides a natural introduction to Boolean algebras. Boolean algebras serve to unify some important aspects of set theory and logic. The early introduction also provides a nontrivial example of an axiomatic system. This example can then be recalled when the axiomatic system is more formally introduced in chapter 3.

The chapter also contains brief sections on informal logic and analyzing claims. Both sections are optional.

Chapter 3: Proof

Chapter 3 provides a careful introduction to proof. The chapter starts with a discussion of axiomatic mathematics. This provides the student with information about the context in which proofs exist. It is necessary to have some content in order to give examples of various proof strategies and provide exercises for practice. This is accomplished by introducing much of the standard material from elementary number theory. This introduction also fills in some of the gaps in the student's background knowledge.

The chapter contains a discussion of the major proof strategies and also has a section that provides hints and suggestions for creating proofs. There is also a careful introduction to mathematical induction.

Chapter 4: Algorithms

Chapter 4 is about algorithms. The two major topics are: expressing algorithms and measuring algorithm efficiency. Section 4.1 provides a fairly complete introduction to pseudocode. Courses taught to sophomore computer science majors can either skip this section or else do a quick review. I have found that students who have not yet taken a programming course really need the detailed descriptions found in this section. As a side benefit, my students tell me that this section was very helpful when they enrolled in a programming course after taking discrete math.

Section 4.2 introduces big-$\mathcal{O}$ and big-Θ. My students tend to vote this material as their least favorite in the course. Since this material, and the ability to apply it, is so important in computer science courses (such as data structures), I have expended extra effort to help the students gain a good intuitive understanding of the basic definitions, the reason those definitions are important, and how to apply them to real algorithms.

The chapter ends with two interesting examples. Section 4.3 compares three different algorithms for solving the same problem (finding a substring in a longer text). They provide an interesting example illustrating the practical difference in finding an algorithm with a better big-Θ reference function. The final algorithm (Boyer-Moore) is also worth studying purely for the cleverness of the ideas that are used.

The short section at the end of the chapter examines a problem for which no algorithm can ever exist: the Halting problem. It also provides a very nice example of a proof by contradiction.

Chapter 5: Counting

Chapter 5 presents the standard material about counting. The notions of independent tasks, mutually exclusive tasks, permutation and combinations (with or without repetition) are all present. In addition, the pigeon-hole principle, inclusion-exclusion, and the multinomial counting theorems are presented. The chapter also contains a section that introduces the notion of a combinatorial proof.

Chapter 8 expands the counting repertoire with a discussion of occupancy problems.

Chapter 6: Finite Probability Theory

Chapter 6 provides the basic definitions and properties of finite probability. It discusses sample spaces, events, independent and mutually exclusive events, and conditional probability. There is a section that applies many of the counting techniques found in chapter 5.

More advance topics include expected value and Bayes' Theorem.

Chapter 7: Recursion

Chapter 7 introduces recursion, first from a computer science perspective (recursive algorithms), and then from a mathematics perspective (recurrence relations). The discussion of recursive algorithms includes numerous nontrivial applications.

Techniques for solving recurrence relations include: back substitution, using the roots of a characteristic equation to solve linear homogeneous recurrence relations with constant coefficients, and the use of generating functions.

There is also a section dedicated to the Master Theorems for finding big-Θ reference functions for divide-and-conquer recursions.

As a bonus, the chapter contains a brief discussion of the Josephus problem. The historical origins are explored, and a simplified version of the problem is solved.

Chapter 8: Combinatorics

Chapter 8 presents a brief overview of some common defining characteristics (existence, enumeration, optimization) of the field of combinatorics. It then explores some sample illustrative topics. The topics include: partitions, occupancy problems and Stirling numbers, Latin squares and finite projective planes, balanced incomplete block designs, the knapsack problem, error-correcting codes, and systems of distinct representatives and Ramsey numbers.

Much of the material in this chapter will stretch typical sophomores. The easier sections are 8.1 (partitions, occupancy problems, and Stirling numbers), 8.3 (balanced incomplete block designs), and 8.4 the knapsack problem. However, section 8.3 does make occasional references to material in section 8.2.

Chapter 9: Formal Models in Computer Science

Whereas Chapter 8 is oriented towards math majors, this chapter is mainly oriented towards computer science majors. The chapter begins with a mathematically motivated derivation of Shannon's mathematical model of *information*. It also contains his familiar model of communication. The material in this initial section is not needed for subsequent sections and so can be omitted without any break in continuity. I have placed it first in the hope that these models will gain more exposure in discrete mathematics courses.

The chapter continues with discussions of finite-state machines and finite automata. Formal languages are introduced next, with most of the discussion centered on regular grammars. A fairly detailed discussion of regular expressions is presented next. The notation is from the standard Unix/perl conventions.

Section 9.5 presents Kleene's theorem and the equivalence of finite automata, regular sets, regular expressions, and regular grammars. Nondeterministic finite automata are introduced in an optional section that contains the proof of Kleene's Theorem.

The chapter concludes with a brief introduction to the Chomsky hierarchy of grammars, pushdown automata, Turing machines, and the Church-Turing Thesis.

Chapter 10: Graphs

Chapter 10 is a fairly lengthy introduction to graph theory. The chapter introduces the basic terminology, numerous examples of graphs and graph families, and the standard material on connectivity and adjacency. Euler circuits and Hamilton cycles are explored, as well as alternative mechanisms for representing graphs in a computer. The notion of graph isomorphism is also discussed. Weighted graphs and Dijkstra's shortest path algorithm are also presented.

The chapter also contains a section that presents four of the most famous theorems in graph theory: Euler's formula, the characterization of regular polyhedra, Kuratowski's Theorem, and the four color theorem. A simple Java applet/application is available for click-and-drag exploration of planarity. The section also contains a proof of the five color theorem.

Chapter 11: Trees

There is sufficient material about trees to warrant a separate chapter. The chapter starts with the standard definitions (root, leaf, balanced, and so on), and some of the basic counting theorems for trees. The notions of tree traversal and searching and sorting are also presented. Section 11.3 contains three interesting applications of trees: parse trees, Huffman compression, and XML. I typically present just one of the three in any given semester. The chapter ends with spanning trees.

Chapter 12: Functions, Relations, Databases, and Circuits

The material in chapter 12 is connected by the concepts of functions and relations. The first two sections present the foundational ideas and the final three sections provide nontrivial applications.

The foundational ideas include the definitions of *function* and *relation* as well as the standard properties that relations might exhibit (reflexive, symmetric, transitive, and so on). There is also a discussion of equivalence relations and a discussion of what it means for a binary operator on a set of equivalence classes to be well-defined.

Much of the material in Sections 12.1 and 12.2 is often presented in the early chapters of other discrete math texts. I have placed it at the end in order to take advantage of some of the topics presented elsewhere in the book (such as the counting techniques in Chapter 5). I also wanted to keep this material close to the applications in Sections 12.3, 12.4, and 12.5.

The nontrivial applications are: a discussion of normal forms in relational databases, binary functions and disjunctive normal form, and the design of combinatorial circuits (including the minimization of binary expressions via the Quine-McCluskey algorithm).

For a one-semester course, I would recommend covering Sections 12.1 and 12.2 and then presenting either Section 12.3 or Sections 12.4 and 12.5.

Appendices

There are several appendices. Appendix A contains a brief review of the standard number systems. Appendix B contains a very brief review of summation notation and Appendix E contains a short introduction to some matrix terminology and arithmetic. Appendix C contains some logic puzzles and is intended as a supplement to Chapter 2. Appendix D contains some background information on the Golden Ratio. Appendix F contains a summary of the Greek alphabet.

Finally, Appendix G will be the most frequently used appendix. It contains detailed solutions to selected Exercises. Consequently, it serves as a free student solution manual. By placing it in the text, students gain in convenience. Of greater importance, instructors can be assured that solutions to problems that are not in Appendix G are not available for purchase by their students. There is no other student solution manual.

SAMPLE COURSES

This text is suitable for courses taught within a math department for mathematics majors, for courses taught within a computer science department for computer science majors, and for courses with bimodal populations. Sample one-semester courses are shown for each of those options. There is also sufficient material in the text to fill a leisurely two-semester course (using one class period for most sections in the text and two class sessions for a few others).

Bimodal Course

The course I teach consists of approximately 45% mathematics and mathematics with secondary licensure majors and 45% computer science majors. The remaining 10% are elementary education majors with a mathematics specialization and physics, engineering, and chemistry majors. The outline below shows the sections I typically cover. You may wish to be a bit less aggressive and delete a few sections.

Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7
1.1–1.4	2.1, 2.3, 2.4, 2.5, 2.6	3.1–3.4	4.1–4.4	5.1–5.3	6.1–6.3 or Ch 8 selections	7.1–7.2

Ch 8	Ch 9	Ch 10	Ch 11	Ch 12
8.1–8.3 or 8.1, 8,3, 8,4 or Ch 6 selections	9.2–9.5	10.1–10.5	11.1, 11.2, one example in 11.3, 11.4	

Mathematics Majors

The sections listed here are appropriate for a course populated with mainly mathematics majors and minors. The list needs to be trimmed a bit to fit in a semester.

Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6
1.1–1.4	2.1–2.6	3.1–3.4	4.1–4.4	5.1–5.3	6.1–6.5

Ch 7	Ch 8	Ch 9	Ch 10	Ch 11	Ch 12
7.1, 7.2, 7.4, 7.5	8.1–8.6	9.1	10.1–10.6	11.1, 11.4	12.1, 12.2, 12.4

Computer Science Majors

The sections listed here are appropriate for a course populated with mainly computer science majors and minors. The list needs to be trimmed a bit to fit in a semester.

Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6
1.1–1.4	2.1, 2.3, 2.4–2.6	3.1–3.4	4.2–4.4	5.1, 5.3	6.1–6.3

Ch 7	Ch 8	Ch 9	Ch 10	Ch 11	Ch 12
7.1–7.3	8.5	9.2–9.6	10.1–10.6	11.1–11.3	12.1, 12.2 and either 12.4 or 12.4, 12.5

ACKNOWLEDGMENTS

Michelle Nielsen, one of my students, has been invaluable as a proof reader. Her meticulous care for details and her perspective from a student's point of view have both contributed towards making this a much better book than it might have been. She has also checked the solutions to every exercise (and even contributed a few exercises of her own). I can't imagine anyone who could have done better.

I want to thank my son, Nathan Gossett, for writing the code for the Sandbox and Quine-McCluskey applets/applications. My wonderful wife, Florence Kuofang Gossett, has been very supportive of this project, as has my daughter, Rachel.

I have appreciated the feedback from the external reviewers. I especially appreciate their care in examining the manuscript, their astute observations and suggestions, and the spirit in which they approached their reviews. The following people reviewed various portions of the manuscript:

Miklos Bona,	University of Florida
William G. Brown,	McGill University
Sheldon Eisenberg,	University of Hartford
Anthony B. Evans,	Wright State University
Timothy J. Ford,	Florida Atlantic University
Johannes H. Hattingh,	Georgia State University
Arthur M. Hobbs,	Texas A&M University
Akihiro Kanamori,	Boston University
Jay Kappraff,	New Jersey Institute of Technology
Panos E. Livadas,	University of Florida
Timothy S. Norfolk,	The University of Akron
Stanley E. Payne,	University of Colorado at Denver
Michael J. Schlosser,	The Ohio State University

I would also like to thank Prentice Hall for choosing to publish this book. In particular, I wish to thank George Lobell, the acquisitions editor, who was willing to support this project when it was in its early childhood.

Many other people at Prentice Hall (in production, marketing, sales, printing, and other departments) have contributed to this project. Four people, in particular, have been

very directly involved with me to complete the final stages of this project. The first is the copy editor, Patricia M. Daly. She has read every page of the original manuscript and has made valuable suggestions regarding the proper way to format the text and has also made needed corrections where my technical writing skills were lacking. The second person is the LaTeX compositor, Dennis Kletzing. He was responsible for incorporating revisions and also for taking my LaTeX source files and modifying them, using macros he created, to conform to the formatting specifications for this project. Next, the A/V editor, Thomas Benfatti, was responsible for ensuring that the figures and other art work in the book were up to publishable standards. Finally, the production editor, Jeanne Audino, and her staff were busy during this phase of the project managing the communication between all the people who were working concurrently on the manuscript, as well as coordinating all of the revisions and scheduling. She kept the work flowing and on schedule.

If you have read this far, you should have realized that there were many people besides the author who contributed significantly to the project. As the author, I am privileged to have my name on the front cover and I also have the duty to take final responsibility for the contents of the book. However, many very competent professionals joined together to convert the original manuscript into something much better: a professionally published textbook. I am deeply grateful for all of these people. Thank you.

I received help and good advice from many people. However, the final content of this text is a direct product of my choices. Consequently, any remaining mistakes are mine. In order to keep this text as accurate as possible, I am offering a reward of $1.00 to the first person who informs me of each mistake or misprint in the book. Mistakes can be reported by using a web browser to look at:

http://www.mathcs.bethel.edu/~gossett/discreteMathWithProof/

and selecting the *Errors* link.

Eric Gossett
Bethel College
3900 Bethel Drive
St. Paul, MN 55112
gossett@bethel.edu

To the Student

Before plunging into the details of this book, it is helpful to do a brief survey of the entire book. A scan through the table of contents will orient you towards some of the topics (mysterious as they may seem before you study them), and familiarize you with some of the other features of the book. In particular, notice the index. A serious attempt has been made to make the index complete and useful. It will also be helpful to do a quick scan of the appendices.

You should also notice that every chapter has a section named "Quick Check Solutions", followed by one named "Chapter Review".

The Chapter Review typically contains a brief summary of the main ideas in the chapter, a list of new notation, lists of the definitions and theorems introduced in the chapter, some sample exam questions (with solutions), and a short collection of projects.

Expectations

My goal in writing this textbook is to equip you, the student, to be able to actually apply the material. This has influenced my attitude towards both exposition and homework exercises. Unlike your high school texts and probably your calculus text, this book is not merely a collection of prototype examples to copy and submit as solutions to homework exercises. The subject matter for this course will require you to change your approach to learning mathematics. There are still numerous examples, but they often illustrate concepts rather than procedures. As such, the examples do not often provide simple templates for solving homework exercises. In fact, many of the homework exercises ask you to create proofs or to solve problems that do not have clone cousins among the examples.

The level of precision that will be expected from you will also increase. This is essential for mathematics majors who will eventually encounter courses in algebraic structures and real analysis. It is also essential for computer science majors who need to develop the ability to create precise and correct algorithms and express them as computer programs.

Reading

Reading a mathematics text requires a different approach than reading other kinds of material. You need to read actively. Have paper and pencil ready. At the end of each sentence or paragraph, ask yourself if you could reproduce the ideas just presented. Don't assume you understand an example until you can reproduce the solution with the book closed. At various places, the reading is interrupted by Quick Check exercises. Quick Check problems provide a chance for you to make sure you understand the current material before reading any further. To receive the greatest benefit, you should write out complete solutions. After (and only after) you have completed your solution, you should compare your solution to the detailed solution at the end of the chapter. Don't proceed with the reading until you understand the textbook solution.

Exercises

Mathematics cannot be learned passively, so it is necessary to provide opportunities for you to practice the concepts presented. However, I do not see much value in extensive sets of drill exercises. In fact, I find that large collections of "clone exercises" tend