Second Edition

Discrete Mathematics for Computer Scientists and Mathematicians

Joe L. Mott

Abraham Kandel

Theodore P. Baker

Second Edition

Discrete Mathematics for Computer Scientists and Mathematicians

Joe L. Mott
Abraham Kandel
Theodore P. Baker
The Florida State University
Department of Mathematics and Computer Science

A Reston Book
Prentice-Hall
Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data

Mott, Joe L.
Discrete mathematics for computer scientists and mathematicians.

Rev. ed. of: Discrete mathematics for computer scientists. c1983.
Bibliography: p.
Includes index.
1. Combinatorial analysis. 2. Graph theory.
3. Mathematics—Data processing. I. Kandel, Abraham.
II. Baker, Theodore P. III. Mott, Joe L. Discrete mathematics for computer scientists. IV. Title.
QA164.M68 1986 555′.6 85-14365
ISBN 0-8359-1391-0

A Reston Book
Published by Prentice-Hall
A Division of Simon & Schuster, Inc.
Englewood Cliffs, NJ 07632

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Preface

This text is intended for use in a first course in discrete mathematics in an undergraduate computer science and mathematics curriculum. The level is appropriate for a sophomore or junior course, and the number of topics and the depth of analysis can be adjusted to fit a one-term or a two-term course. A computer science student can take this course concurrently with the first course in programming preliminary to the study of data structures and the design and analysis of algorithms. A mathematics student may take this course concurrently with the first calculus course.

No specific background is prerequisite outside of the material ordinarily covered in most college algebra courses. In particular, a calculus background is not required for Chapters 1 to 7. While it is not necessary, knowledge of limits would help in understanding the proof of one theorem in Chapter 7 and knowledge of integration would enhance understanding some of the discussions in Chapter 8. We have assumed that students will have had little or no programming experience, although it would be desirable.

Our assumption about background has dictated how we have written the text in certain places. For instance, in Chapter 3, we have avoided reference to the convergence of power series by representing the geometric series

$$\sum_{i=0}^{\infty} a^i X^i$$

as the multiplicative inverse of $1 - aX$; in other words, we have considered power series from a strictly algebraic rather than the analytical viewpoint. Likewise, in Chapter 4, we avoid reference to limits when we discuss the asymptotic behavior of functions and the "big O notation," but if students understand limits, then exercises 11 and 12 in Section 4.2.1 will greatly streamline the discussion.

The Association for Computing Machinery, CUPM, and others have recommended that a computer science curriculum include a discrete mathematics course that introduces the student to logical and algebraic structures and to combinatorial mathematics including enumeration methods and graph theory. This text is an attempt to satisfy that recommendation.

Furthermore, we expect that some of the teachers of this course will be mathematicians who are not computer scientists by profession or by training. Therefore, we have purposely suppressed writing many algorithms in computer programming language, although on occasions it would have been easier to do so.

We believe that a discrete mathematics course based on our book will meet several important needs of both computer science and mathematics majors. While the basic content of the book is mathematics, many applications are oriented toward computer science. Moreover, we have attempted to include examples from computer science that can be discussed without making presumptions about the reader's background in computer science.

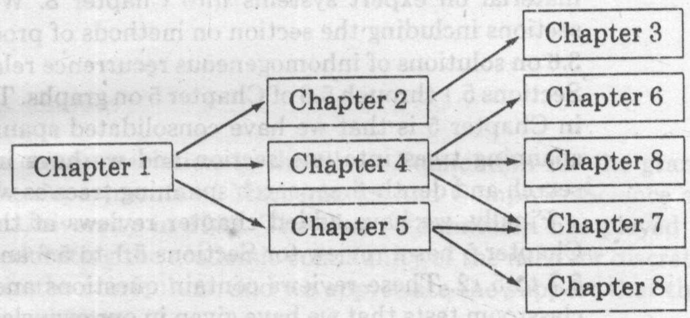
Many apparently mathematical topics are quite useful for computer science students as well. In particular, computer science students need to understand graph theory, since many topics of graph theory will be applied in a data structures course. Moreover, they need mathematical induction as a proof technique and to understand recursion, Boolean algebra to prepare for digital circuit design, logic and other proof techniques to be able to prove correctness of algorithms, and recurrence relations to analyze algorithms. Besides that, computer science students need to see how some real life problems can be modeled with graphs (like minimal spanning trees in Section 5.4, scheduling problems and graph coloring in Section 5.11, and network flow problems in Chapter 7).

Mathematics majors, on the other hand, will use graphs as a modeling tool, and they will benefit from a study of recurrence relations to understand computer solutions of differential equations. But more than that, discrete mathematics provides a good training ground for the mathematics student to learn to solve problems and to make correct proofs. For this reason, mathematics majors should take discrete mathematics quite early in their program of studies, preferably before those courses that require many proofs.

Discrete mathematics embodies the spirit of mathematical and scientific research perhaps more than almost any other undergraduate mathematics course. In graph theory, for example, powerful concepts can be defined and grasped because they can be visualized and simple examples can be constructed easily. This feature and others make the subject both challenging and rewarding to student and teacher alike.

The text has evolved over a period of years and, in that time, our curriculum at Florida State University has changed significantly, especially for computer science students. Thus, not only has the list of topics changed, but also the order in which we discuss them. Consequently, we have written the text so that the chapters are more or less independent of each other.

The following diagram shows the basic logical relationship among the chapters.



Chapter 1, of course, is introductory and as much or as little of it can be discussed as needed depending on the background of the students. Most students likely have been exposed to the material of Sections 1.1, 1.2, and 1.3 except possibly the definitions in Section 1.3 of equivalence relations, composition of relations, and one-to-one and onto functions.

We recommend covering, at the minimum, Section 1.7 (Methods of Proof of an Implication) and Section 1.10 (Induction). Sections 1.5 and 1.6 contain introductory material on logic and is the foundation upon which Section 1.7 is built. A thorough understanding of proof by induction is, in our opinion, absolutely essential.

Section 1.4 is a general discussion that can be assigned for reading. Section 1.9 (Rules of Inference for Quantified Propositions) may be omitted without injury.

Chapter 3 can be taught at any time after Chapter 2 is covered. In particular, in a curriculum that calls for an early introduction to trees and graph theory we recommend that Chapter 3 be postponed until after Chapter 5. Only elementary recurrences are used in Section 5.5, and in Section 5.6 there is only one use of a recurrence relation. But even this does not require any result from Chapter 3, as a solution can be obtained instead from Example 1.10.11 in Chapter 1.

Chapter 4 on directed graphs and Chapter 5 on nondirected graphs are related but may be treated as mutually independent chapters since definitions given in Chapter 4 for digraphs are repeated and illustrated for Chapter 5. In fact, Sections 4.1 and 4.2 can be taught concurrently with Sections 5.1 and 5.2.

We have made several significant changes from the first edition. First we have added two chapters, Chapter 7 on network flows and Chapter 8 on representation and manipulation of imprecision. Next, we have added several exercises in almost every section of the book. Moreover, we have consolidated two separate sections on partial orders into one in this second edition (Section 4.4), and we have removed the material on fuzzy sets from Chapter 1 of the first edition and incorporated that with other material on expert systems into Chapter 8. We have rewritten other sections including the section on methods of proof in Chapter 1, Section 3.6 on solutions of inhomogeneous recurrence relations in Chapter 3, and Sections 5.1 through 5.6 of Chapter 5 on graphs. The most notable change in Chapter 5 is that we have consolidated spanning trees and minimal spanning trees into one section and we have introduced breadth-first search and depth-first search spanning trees as well.

Finally, we have added chapter reviews at the end of each chapter. Chapter 5 has a review for Sections 5.1 to 5.6 and then one for Sections 5.7 to 5.12. These reviews contain questions and problems from actual classroom tests that we have given in our own classes.

There are several possible course syllabi. For mathematics students only, we suggest Chapters 1, 2, 3, 5, and 7. One for computer science majors alone could be Chapters 1, 2, 4, 5 (at least Sections 5.1 to 5.6), 7, and 8. Chapter 6 on Boolean algebras could replace Chapter 7 or 8 if preparation for a digital design course is needed.

At Florida State University our discrete classes contain both mathematics majors and computer science majors so we follow this syllabus:

Discrete I:

Sections 1.5 to 1.10 of Chapter 1 (Section 1.9 is optional), Chapter 2, and Chapter 5 (at least Sections 5.1 to 5.6)

Discrete II:

Chapters 3, 4, 7, and selected topics from sections 5.7 to 5.12 as time permits.

Exercises follow each section, and as a general rule the level of difficulty ranges from the routine to the moderately difficult, although some proofs may present a challenge. In the early chapters we include many worked-out examples and solutions to the exercises hoping to enable the student to check his work and gain confidence. Later in the book we make greater demands on the student; in particular, we expect the student to be able to make some proofs by the end of the text.

Acknowledgments

We express our appreciation to the Sloan Foundation for the grant awarded to the departments of Mathematics and Computer Science at Florida State University in 1983. The Sloan Foundation has played a major role in educating the academic community of the need for discrete mathematics in the curriculum, and we appreciate the support that the Foundation has given us.

To our colleagues and friends who have taught from an earlier version of the book and made suggestions for improvement we say a heartfelt thank you.

The editorial staff at Reston Publishing have been a great help and we thank them.

Portions of the material in Chapter 8 are based on recent work by Lofti A. Zadeh [50], Maria Zemankova-Leech and Abraham Kandel [52], L. Applebaum and E. H. Ruspini [46], and many researchers in the fields of fuzzy set theory and artificial intelligence. Special thanks are due Dalya and Peli Pelled, who provided the desk upon which A. Kandel wrote Chapter 8.

We wish to express our gratitude to several people who helped with the preparation of the manuscript. Sheila O'Connell and Pam Flowers read early versions and made several helpful suggestions while Sandy Robbins, Denise Khosrow, Lynne Pennock, Ruth Wright, Karen Serra, and Marlene Walker typed portions of the manuscript for the first edition. Robert Stephens typed most of the manuscript for the second edition.

Finally, we want to express our love and appreciation to our families for their patience and encouragement throughout the time we were writing this book.

A Note to the Reader

In each chapter of this book, sections are numbered by chapter and then section. Thus, section number 4.2 means that it is the second section of Chapter 4. Likewise theorems, corollaries, definitions, and examples are numbered by chapter, section, and sequence so that example 4.2.7 means that the example is the seventh example in section 4.2.

The end of every theorem proof is indicated by the symbol □.

We acknowledge our intellectual debt to several authors. We have included at the end of the book a bibliography which references many, but not all, of the books that have been a great help to us. A bracket, for instance [25], means that we are referring to the article or book number 25 in the bibliography.

An asterisk (*) indicates that the problem beside which the asterisk appears is generally more difficult than the other problems of the section.

Joe L. Mott
Abraham Kandel
Theodore P. Baker

Contents

1

Foundations

1.1 BASICS

One of the important tools in modern mathematics is the theory of sets. The notation, terminology, and concepts of set theory are helpful in studying any branch of mathematics. Every branch of mathematics can be considered as a study of sets of objects of one kind or another. For example, algebra is concerned with sets of numbers and operations on those sets whereas analysis deals mainly with sets of functions. The study of sets and their use in the foundations of mathematics was begun in the latter part of the nineteenth century by Georg Cantor (1845–1918). Since then, set theory has unified many seemingly disconnected ideas. It has helped to reduce many mathematical concepts to their logical foundations in an elegant and systematic way and helped to clarify the relationship between mathematics and philosophy.

What do the following have in common?

- a crowd of people,
- a herd of animals,
- a bunch of flowers, and
- a group of children.

In each case we are dealing with a collection of objects of a certain type. Rather than use a different word for each type of collection, it is convenient to denote them all by the one word "set." Thus a **set** is a collection of well-defined objects, called the **elements** of the set. The elements (or **members**) of the set are said to belong to (or be contained in) the set.

It is important to realize that a set may itself be an element of some other set. For example, a line is a set of points; the set of all lines in the plane is a set of sets of points. In fact a set can be a set of sets of sets and so on. The theory dealing with the (abstract) sets defined in the above manner is called **(abstract or conventional) set theory,** in contrast to fuzzy set theory which will be introduced later in Chapter 8.

This chapter begins with a review of set theory which includes the introduction of several important classes of sets and their properties.

In this chapter we also introduce the basic concepts of relations and functions necessary for understanding the remainder of the material. The chapter also describes different methods of proof—including mathematical induction—and shows how to use these techniques in proving results related to the content of the text.

The material in Chapters 2–8 represents the applications of the concepts introduced in this chapter. Understanding these concepts and their potential applications is good preparation for most computer science and mathematics majors.

1.2 SETS AND OPERATIONS OF SETS

Sets will be denoted by *capital* letters $A,B,C,\ldots,X,Y,Z$. Elements will be denoted by *lower case* letters $a,b,c,\ldots,x,y,z$. The phrase "is an element of" will be denoted by the symbol $\in$. Thus we write $x \in A$ for "x is an element of A." In analogous situations, we write $x \notin A$ for "x is not an element of A."

There are five ways used to describe a set.

1. Describe a set by describing the properties of the members of the set.
2. Describe a set by listing its elements.
3. Describe a set A by its characteristic function, defined as

$$\mu_A(x) = 1 \text{ if } x \in A,$$

$$\mu_A(x) = 0 \text{ if } x \notin A,$$

for all x in U, where U is the universal set, sometimes called the "universe of discourse," or just the "universe," which is a fixed specified set describing the context for the duration of the discussion.

If the discussion refers to dogs only, for example, then the universe of discourse is the class of dogs. In elementary algebra or number theory, the universe of discourse could be numbers (rational, real, complex, etc.). The universe of discourse must be explicitly stated, because the truth value of a statement depends upon it, as we shall see later.

4. Describe a set by a recursive formula. This is to give one or more elements of the set and a rule by which the rest of the elements of the set may be generated. We return to this idea in Section 1.10 and in Chapter 3.

5. Describe a set by an operation (such as union, intersection, complement, etc.) on some other sets.

Example 1.2.1. Describe the set containing all the nonnegative integers less than or equal to 5.

Let A denote the set. Then the set A can be described in the following ways:

1. $A = \{x \mid x \text{ is a nonnegative integer less than or equal to } 5\}$.
2. $A = \{0,1,2,3,4,5\}$.
3. $\mu_A(x) = \begin{cases} 1 \text{ for } x = 0,1,\ldots,5, \\ 0 \text{ otherwise.} \end{cases}$
4. $A = \{x_{i+1} = x_i + 1, i = 0,1,\ldots,4, \text{ where } x_0 = 0\}$.
5. This part is left to the reader as an exercise to be completed once the operations on sets are discussed.

The use of braces and | ("such that") is a conventional notation which reads: $\{x \mid \text{property of } x\}$ means "the set of all elements x *such that* x has the given property." Note that, for a given set, not all the five ways of describing it are always possible. For example, the set of real numbers between 0 and 1 cannot be described by either listing all its elements or by a recursive formula.

In this section, we shall introduce the fundamental operations on sets and the relations among these operations. We begin with the following definitions.

Definition 1.2.1. Let A and B be two sets. Then A is said to be a **subset** of B if every element of A is an element of B; A is said to be a proper subset of B if A is a subset of B and there is at least one element of B which is not in A.

If A is a subset of B, we say A is contained in B. Symbolically, we write $A \subseteq B$. If A is a proper subset of B, then we say A is strictly contained in

B, denoted by $A \subset B$. The containment of sets has the following properties. Let A, B, and C be sets.

1. $A \subseteq A$.
2. If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.
3. If $A \subset B$ and $B \subset C$, then $A \subset C$.
4. If $A \subseteq B$ and $A \not\subseteq C$, then $B \not\subseteq C$, where $\not\subseteq$ means "is not contained in."

The statement $A \subseteq B$ does not rule out the possibility that $B \subseteq A$. In fact, we have both $A \subseteq B$ and $B \subseteq A$ if and only if (abbreviated iff) A and B have the same elements. Thus we define the following:

Definition 1.2.2. Two sets A and B are equal iff $A \subseteq B$ and $B \subseteq A$. We write $A = B$.

Therefore, we have the following principle.

Principle. To show that two sets A and B are equal, we must show that each element of A is also an element of B, and conversely.

A set containing no elements is called the **empty set** or **null set,** denoted by $\varnothing$. For example, given the universal set U of all positive numbers, the set of all positive numbers x in U satisfying the equation $x + 1 = 0$ is an empty set since there are no positive numbers which can satisfy this equation. The empty set is a subset of every set. In other words, $\varnothing \subseteq A$ for every A. This is because there are no elements in $\varnothing$; therefore, every element in $\varnothing$ belongs to A. It is important to note that the sets $\varnothing$ and $\{\varnothing\}$ are very different sets. The former has no elements, whereas the latter has the unique element $\varnothing$. A set containing a single element is called a **singleton.**

We shall now describe three operations on sets; namely, complement, union, and intersection. These operations allow us to construct new sets from given sets. We shall also study the relationships among these operations.

Definition 1.2.3. Let U be the universal set and let A be any subset of U. The **absolute complement** of A, $\overline{A}$, is defined as $\{x \mid x \notin A\}$ or, $\{x \mid x \in U \text{ and } x \notin A\}$. If A and B are sets, the **relative complement** of A with respect to B is as shown below.

$$B - A = \{x \mid x \in B \text{ and } x \notin A\}.$$

It is clear that $\overline{\varnothing} = U$, $\overline{U} = \varnothing$, and that the complement of the complement of A is equal to A.

Definition 1.2.4. Let A and B be two sets. The **union** of A and B is $A \cup B = \{x \mid x \in A \text{ or } x \in B \text{ or both}\}$. More generally, if $A_1, A_2, \ldots, A_n$ are sets, then their union is the set of all objects which belong to at least one of them, and is denoted by

$$A_1 \cup A_2 \cup \cdots \cup A_n, \text{ or by } \bigcup_{j=1}^{n} A_j.$$

Definition 1.2.5. The **intersection** of two sets A and B is $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$. The intersection of n sets $A_1, A_2, \ldots, A_n$ is the set of all objects which belong to every one of them, and is denoted by

$$A_1 \cap A_2 \cap \cdots \cap A_n, \text{ or } \bigcap_{j=1}^{n} A_j.$$

Some basic properties of union and intersection of two sets are as follows:

	Union	Intersection
Idempotent:	$A \cup A = A$	$A \cap A = A$
Commutative:	$A \cup B = B \cup A$	$A \cap B = B \cap A$
Associative:	$A \cup (B \cup C) = (A \cup B) \cup C$	$A \cap (B \cap C) = (A \cap B) \cap C$

It should be noted that, in general,

$$(A \cup B) \cap C \neq A \cup (B \cap C).$$

Definition 1.2.6. The **symmetrical difference** of two sets A and B is $A \triangle B = \{x \mid x \in A, \text{ or } x \in B, \text{ but not both}\}$. The symmetrical difference of two sets is also called the **Boolean sum** of the two sets.

Definition 1.2.7. Two sets A and B are said to be **disjoint** if they do not have a member in common, that is to say, if $A \cap B = \varnothing$.

We can easily show the following theorems from the definitions of union, intersection, and complement.

Theorem 1.2.1. (Distributive Laws). Let A, B, and C be three sets. Then,

$$C \cap (A \cup B) = (C \cap A) \cup (C \cap B),$$
$$C \cup (A \cap B) = (C \cup A) \cap (C \cup B).$$