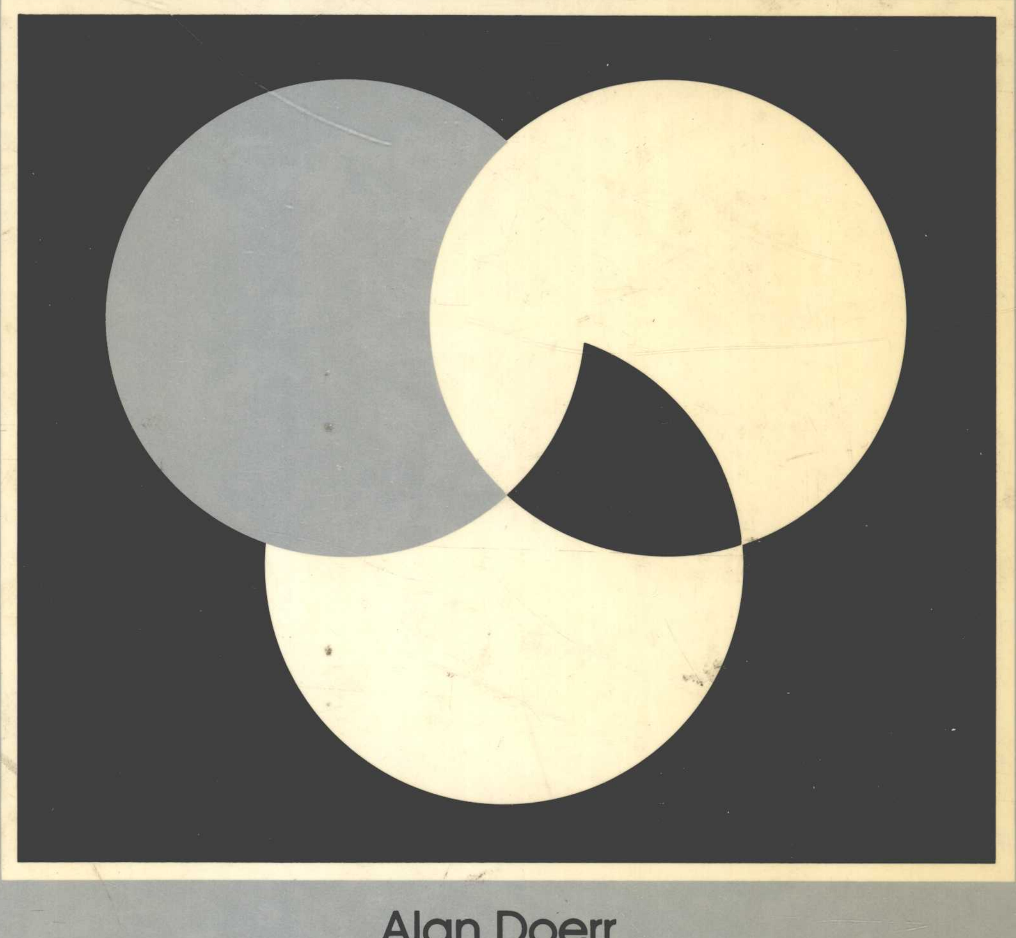
APPLIED DISCRETE STRUCTURES FOR COMPUTER SCIENCE



Alan Doerr
Kenneth Levasseur

APPLIED DISCRETE STRUCTURES for COMPUTER SCIENCE

Alan Doerr

Kenneth Levasseur

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF LOWELL

SRA®

SCIENCE RESEARCH ASSOCIATES, INC.
Chicago, Henley-on-Thames, Sydney, Toronto

A Subsidiary of IBM

Acquisition Editor	Alan W. Lowe
Project Editor	Mary C. Konstant
Compositor	Interactive Composition Corp.
Illustrator	Alex Teshin
Cover and Text Designer	Carol Harris

The SRA Computer Science Series

William A. Barrett and John D. Couch, *Compiler Construction: Theory and Practice*
Marilyn Bohl and Arline Walter, *Introduction to PL/1 Programming and PL/C*
Alan W. Doerr and Kenneth M. Levasseur, *Applied Discrete Structures for Computer Science*
Mark Elson, *Concepts of Programming Languages*
Mark Elson, *Data Structures*
Peter Freeman, *Software Systems Principles: A Survey*
Philip Gilbert, *Software Design and Development*
A. N. Habermann, *Introduction to Operating System Design*
Harry Katzan, Jr., *Computer Systems Organization and Programming*
Henry Ledgard and Michael Marcotty, *The Programming Language Landscape*
Stephen M. Pizer, *Numerical Computing and Mathematical Analysis*
Harold S. Stone, *Introduction to Computer Architecture, Second Edition*
Gregory F. Wetzel and William G. Bulgren, *The Algorithmic Process: An Introduction to Problem Solving*

Library of Congress Cataloging in Publication Data

Doerr, Alan W., 1938–
Applied discrete structures for computer science.

Bibliography: p.
Includes index.
1. Electronic data processing—Mathematics.
I. Levasseur, Kenneth M., 1950– . II. Title.
QA76.9.M35D64 1985 511 84-23491
ISBN 0-574-21755-X

Printed in the United States of America.

10 9 8 7 6 5 4 3 2

APPLIED DISCRETE STRUCTURES for COMPUTER SCIENCE

To our families

Donna, Christopher, Melissa, and Patrick Doerr

and

Karen, Joseph, Kathryn, and Matthew Levasseur

Preface

This book in discrete mathematics is intended to supply the typical freshman or sophomore in computer science and related disciplines with a first exposure to the mathematical topics essential to their study of computer science or digital logic. It also provides students who are preparing for an advanced-degree program with the background necessary for further study in theoretical computer science. It can be used in either a one- or a two-semester course. Written for the student, this text is the synthesis of many years of experience in teaching this and related courses to students at all undergraduate levels. It offers a unified treatment of the material outlined in all current national recommendations on discrete methods and applied algebra. A major feature of this text is its versatility. Sufficient topics have been included to accommodate students with varied backgrounds.

Chapter Coverage

Chapters 1 and 2 cover elementary concepts in sets and combinatorics. We have found that, although most, if not all, authors assume knowledge of these topics, the typical student needs considerable exposure to them. Applications relevant to computer science students are initiated in these first chapters.

Chapter 3, on logic, lays the framework for all subsequent material. The detail with which logic is developed stresses its overwhelming importance.

Chapter 4 expands on the first chapter on set theory. It utilizes the chapter on logic to further develop concepts in proofs and is an initiation to the topic of algebraic systems.

Students may find some of the topics in Chapters 1 through 4 somewhat theoretical in nature. So Chapter 5, "Introduction to Matrix Algebra," gives them a necessary "breather" from these theoretical concepts. Although some authors utilize matrix algebra in graph theory, none of them review this topic. We have found that many students are completely unfamiliar with matrix algebra.

Chapters 6 and 7 introduce the student to the concepts of relations and functions. By studying the numerous examples, students will become comfortable with these topics, which are crucial to their understanding of the remainder of this text.

Chapter 8 begins with a discussion of how recursion appears in algorithms, definitions, functions, proofs, etc. Major applications are recurrence relations and generating functions. Numerous examples of a variety of recurrence relations are presented in considerable detail in Sections 8.3 and 8.4.

Chapters 9 and 10 are an introduction to graphs and trees. They focus on the description of basic problems involving graphs and trees and their applications.

Chapter 11 is a formalization of concepts of algebraic structures that were introduced in previous chapters. This is done concretely through an introduction to the theory of groups. The chapter culminates in a description of how the idea of an algebraic structure has been adopted to object-oriented computer design.

Chapter 12 is a further development of matrix algebra. The first half of the chapter includes methods for solving systems of equations and how they can be used to compute matrix inverses. The second half is a development of the diagonalization process, including a brief introduction to vector spaces. Applications to recurrence relations and graph theory are given.

In Chapter 13, Boolean algebras are introduced naturally as an algebraic system, motivated by the similarities of logic and set theory. The focus is on examples and illustrations, while theory is explored. Logic design is a culminating application.

Chapter 14 covers the topics of monoids, languages, and finite-state machines and how they are interrelated.

In Chapter 15, we continue our discussion of groups with a further development of the theory and applications that include computations by homomorphic images and coding theory.

Chapter 16 is intended to introduce the student to basic concepts of ring and fields. The key ideas are developed by relying on the student's knowledge of high-school algebra. Polynomials, formal power series, and finite fields are discussed.

Features

Readability. Our students, who we feel are representative of typical undergraduate computer science students, have found all of the texts that we have used difficult to read. We believe that this occurs because typical students lack the background that most authors of discrete mathematics texts assume they have. The chapters on basic set theory, combinatorics, logic, and matrix algebra assist students who are weak in these areas. We have found that time devoted to these topics is well spent and pays off when more abstract topics are covered. Another factor that affects readability is the quantity and quality of examples that are relevant to the material being introduced. By providing numerous, clear examples, we hope that we have made this material more accessible to most students.

Applications. Whenever a major theoretical topic is covered, it is reinforced with at least one application to computer science so that students are able to apply key concepts immediately.

Pascal Notes. Discussions relating to Pascal and other programming languages appear throughout the text, but are clearly marked so that they can be avoided, if desired. In many cases, the Pascal Notes should be understandable to anyone who has had a course in a high-level programming language. We expect that the Pascal Notes will be of use to many students immediately. Some of our own students have commented that the Pascal Notes have affected the way that they write programs.

Coverage. This text is a synthesis of all national guidelines for the discrete methods/applied algebra sequence. Through our applications-oriented, hands-on approach, we feel that these guidelines can be followed without automatically losing a significant percentage of the students because they cannot follow explanations.

Exercises. With the exception of the two opening chapters, exercises immediately follow most sections. Problem sets for the first two chapters appear at the end of the chapter. The problem sets are divided into three sections. Section A consists of problems that all students should be able to do. They are often of a computational nature. Section B consists of a mixture of computational and theoretical problems that the average student should find difficult, but not impossible. Section C consists of challenging problems that suggest extensions of topics appearing in the text or introduce secondary topics.

This book ends with solutions and hints to selected exercises, a table of symbols, a bibliography, and an index.

Suggestions for Classroom Coverage

The material in this book is sufficient to fill two semesters for students who have a reasonable background in algebra. For a one-semester course, the instructor could choose from a variety of options to adapt the material to students' needs. For example:

Chapters 1, 2, 3, 4, 5, 6, 7, and 8:	An introductory one-semester course for the typical student
Chapters 1, 2, 3, 5, 6, 8, 9, and 10:	A non-algebraic introduction for the typical student
Chapters 2, 3, 6, 7, 8, 9, 10, and 13:	A non-algebraic approach for more advanced students
Chapters 4, 6, 7, 8.1, 11, 13, 14, 15, and 16:	An introductory applied algebra course for advanced students

Chapter-by-Chapter Comments

Chapters 1 and 2: These chapters have been written for the student who has no background in sets and elementary combinatorics. Upper-level students can be assigned these chapters for review.

Chapter 3: Nearly all of this chapter is essential, but care must be taken to avoid getting bogged down here. Section 3.5, on mathematical systems, may be covered lightly if proofs will not be emphasized. Section 3.8, on quantifiers, may also be covered lightly if the instructor does not habitually use them.

Chapter 4: Much of this chapter can be skipped if proofs will not be emphasized. There are two exceptions, however. The laws of set theory should be examined and compared to the laws of logic, and the concept of a partition should be discussed. The sections on minsets and duality can be omitted if Boolean algebras are not included in your course.

Chapter 5: This chapter is written for the student who has no background in matrix algebra. For many students, it can be used as a reading assignment.

Chapters 6 and 7: These chapters should be covered in their entirety.

Chapter 8: We feel that the first three sections of this chapter are essential for most students. Unless the focus of the course is algorithmic, Section 8.4 can be completely omitted. We believe that a brief introduction (approximately three lecture hours) to generating functions (Section 8.5) is essential.

Chapter 9: The first two sections of this chapter should be given as a reading assignment with minimal classroom discussion (one lecture hour).

Chapter 10: Classroom coverage of this chapter will depend on the students' previous exposure to trees in a course such as data structures.

Chapter 11: This chapter is essential to the appropriate coverage of Chapters 12 through 16, any of which can be covered independently.

Chapters 12 through 16: In the typical one-semester course, there may be time to cover one of these chapters in detail. Nearly all of these chapters can be covered in a two-semester course. In an applied algebra course, the main concentration would be on these chapters.

Acknowledgments

In preparing this text, the authors have taken advantage of suggestions from a variety of people. We are indebted to our editors, Alan Lowe and Mary Konstant, our colleagues in the mathematics and computer science departments, our students, and a number of anonymous reviewers. We also would like to acknowledge the helpful comments provided by the following reviewers: David C. Buchthal, University of Akron; John Morrison, Towson State University; and Francis L. Schneider, Furman University.

A.W.D.
K.M.L.

Contents