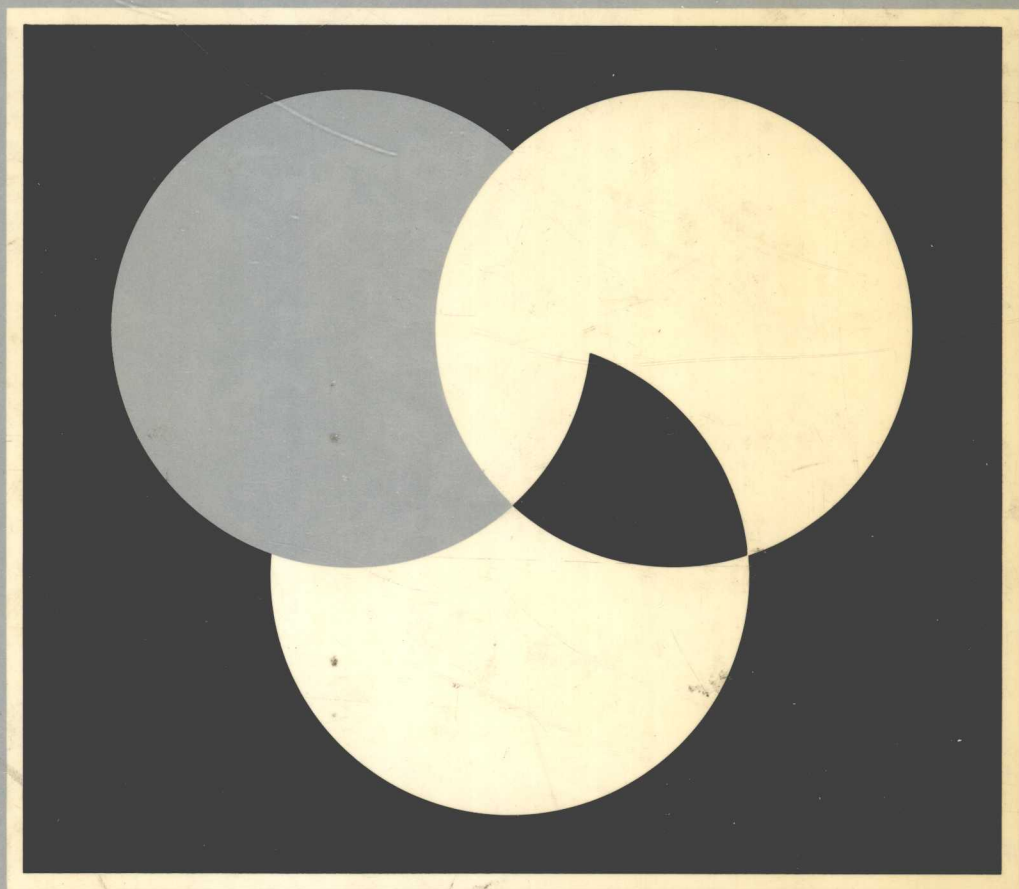
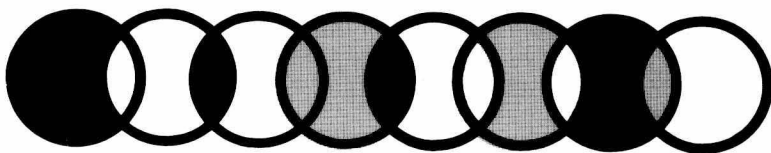


APPLIED DISCRETE STRUCTURES FOR COMPUTER SCIENCE



Alan Doerr
Kenneth Levasseur

APPLIED DISCRETE STRUCTURES for COMPUTER SCIENCE



Alan Doerr

Kenneth Levasseur

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF LOWELL



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

A Subsidiary of IBM

Acquisition Editor
Project Editor
Compositor
Illustrator
Cover and Text Designer

Alan W. Lowe
Mary C. Konstant
Interactive Composition Corp.
Alex Teshin
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/I 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

Copyright © Science Research Associates, Inc. 1985. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Science Research Associates, Inc.

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

Preface	xvii
CHAPTER 1 SET THEORY I	1
Set Notation and Description	2
Subsets	4
Basic Set Operations	4
Venn Diagrams	5
Pascal Note	8
Exercises for Chapter 1	8
CHAPTER 2 COMBINATORICS	11
Rule of Products	13
Permutations	13
Combinations	15
Power Sets	16
Exercises for Chapter 2	17
CHAPTER 3 LOGIC	21
3.1 Propositions and Logical Operators	22
Propositions	22
Logical Operators	22
3.2 Truth Tables and Propositions Generated by a Set	25
Exercises for 3.1 and 3.2	27
3.3 Equivalence and Implication	28
Tautologies	28

Implication	30
Pascal Note	31
Exercises for 3.3	31
3.4 The Laws of Logic	32
Pascal Note	33
Exercises for 3.4	33
3.5 Mathematical Systems	35
Proofs	37
Proofs in Propositional Calculus	37
Direct Proofs	38
Conditional Conclusions	39
Indirect Proofs	40
Proof Style	40
Exercises for 3.5	41
3.6 Propositions Over a Universe	42
Truth Set	42
Pascal Note	44
Exercises for 3.6	44
3.7 Mathematical Induction	45
The Principle of Mathematical Induction	47
Variations	48
Historical Note	50
Exercises for 3.7	50
3.8 Quantifiers	51
The Existential Quantifier	52
The Universal Quantifier	52
The Negation of Quantified Propositions	53
Multiple Quantifiers	53
Exercises for 3.8	55
CHAPTER 4 MORE ON SETS	57
4.1 Methods of Proof for Sets	58
Examples and Counterexamples	58
Proof Using Venn Diagrams	59
Proof Using Set-Membership Tables	59
Proof Using Definitions	61
Exercises for 4.1	63
4.2 Laws of Set Theory	64
Proofs Using Previously Proven Theorems	65

Proof Using the Indirect Method	65
Exercises for 4.2	66
4.3 Partitions of Sets	67
Exercises for 4.3	68
4.4 Minsets	69
Exercises for 4.4	69
4.5 The Duality Principle	72
Exercises for 4.5	72
 CHAPTER 5 INTRODUCTION TO MATRIX ALGEBRA	 73
5.1 Basic Definitions	74
5.2 Addition and Scalar Multiplication	75
5.3 Multiplication of Matrices	76
Exercises for 5.3	78
5.4 Special Types of Matrices	80
Exercises for 5.4	82
5.5 Laws of Matrix Algebra	84
Exercises for 5.5	84
5.6 Matrix Oddities	85
Exercises for 5.6	86
 CHAPTER 6 RELATIONS	 87
6.1 Basic Definitions	88
Relation Notation	89
Exercises for 6.1	90
6.2 Graphs of Relations	91
Exercises for 6.2	95
6.3 Properties of Relations	96
Exercises for 6.3	98
6.4 Matrices of Relations	102
Exercises for 6.4	104
6.5 Closure Operations on Relations	105

Warshall's Algorithm	107
Exercises for 6.5	108
CHAPTER 7 FUNCTIONS	111
7.1 Definition of a Function and Notation	112
Functions of Two Variables	114
Pascal Note	114
Exercises for 7.1	115
7.2 Injective, Surjective, Bijective Functions	116
Exercises for 7.2	118
7.3 Composition, Identity and Inverse	119
Composition	120
Identity Function	121
Inverse Function	122
Exercises for 7.3	124
CHAPTER 8 RECURSION AND RECURRENCE RELATIONS	127
8.1 The Many Faces of Recursion	128
Polynomials and Their Evaluation	128
Pascal Note	131
Recursion	132
Iteration	132
Induction and Recursion	133
Exercises for 8.1	134
8.2 Sequences, or Discrete Functions	134
A Fundamental Problem	136
Exercises for 8.2	136
8.3 Recurrence Relations	138
Solving a Recurrence Relation	138
Recurrence Relations Obtained from "Solutions"	139
Solution of Nonhomogeneous Finite Order	
Linear Relations	144
Base of Right-Hand Side Equal to Characteristic Root	147
Exercises for 8.3	147
8.4 Some Common Recurrence Relations	150
Review of Logarithms	152
Exercises for 8.4	158

8.5	Generating Functions	159
	Solution of a Recurrence Relation Using Generating Functions	160
	Closed Form Expressions for Generating Functions	167
	Extra for Experts	171
	Exercises for 8.5	173
CHAPTER 9	GRAPH THEORY	175
9.1	Graphs—A General Introduction	175
	A Summary of Path Notation and Terminology	179
	Isomorphic Graphs	183
	Exercises for 9.1	183
9.2	Data Structures for Graphs	185
	Exercises for 9.2	192
9.3	Connectivity	193
	Exercises for 9.3	198
9.4	Traversals	199
	The Konigsberg Bridge Problem and Eulerian Graphs	199
	Hamiltonian Graphs	202
	Exercises for 9.4	206
9.5	Graph Optimization	208
	The Traveling Salesman Problem	206
	The Traveling Salesman Problem—Unit Square Version	211
	Networks and the Maximum Flow Problem	213
	Maximal Flows	215
	Other Graph Optimization Problems	218
	Exercises for 9.5	219
9.6	Planarity and Colorings	220
	Graph Coloring	224
	Exercises for 9.6	228
CHAPTER 10	TREES	231
10.1	What is a Tree?	232
	Exercises for 10.1	235
10.2	Spanning Trees	236
	The Minimal Spanning Tree Problem	237

	The Minimum Diameter Spanning Tree Problem	239
	Exercises for 10.2	240
10.3	Rooted Trees	240
	Kruskal's Algorithm	244
	Exercises for 10.3	247
10.4	Binary Trees	249
	Traversals of Binary Trees	250
	Expression Trees	252
	Counting Binary Trees	255
	Exercises for 10.4	256
CHAPTER 11	ALGEBRAIC SYSTEMS	257
11.1	Operations	259
	Common Properties of Operations	259
	Operation Tables	260
	Exercises for 11.1	261
11.2	Algebraic Systems	261
	Levels of Abstraction	263
	Groups	264
	Exercises for 11.2	265
11.3	Some General Properties of Groups	266
	Exercises for 11.3	269
11.4	\mathbf{Z}_n The Integers Modulo n	270
	Properties of Modular Arithmetic on \mathbf{Z}_n	271
	Pascal Note	272
	Exercises for 11.4	272
11.5	Subsystems	274
	Exercises for 11.5	277
11.6	Direct Products	279
	Exercises for 11.6	284
11.7	Isomorphisms	285
	Procedure for Showing Groups are Isomorphic	288
	Procedure for Showing Groups are not Isomorphic	290
	Exercises for 11.7	291

11.8	Object-Oriented Programming	292
	Exercises for 11.8	293
CHAPTER 12	MORE MATRIX ALGEBRA	295
12.1	Systems of Linear Equations	296
	Exercises for 12.1	303
12.2	Matrix Inversion	304
	Exercises for 12.2	307
12.3	An Introduction to Vector Spaces and the Diagonalization Process	308
	Exercises for 12.3	315
12.4	The Diagonalization Process	317
	Exercises for 12.4	324
12.5	Some Applications	326
	Exercises for 12.5	330
CHAPTER 13	BOOLEAN ALGEBRA	333
13.1	Posets Revisited	334
	Exercises for 13.1	336
13.2	Lattices	337
	Exercises for 13.2	341
13.3	Boolean Algebras	342
	Exercises for 13.3	345
13.4	Atoms of a Boolean Algebra	346
	Exercises for 13.4	349
13.5	Finite Boolean Algebras as n-tuples of 0's and 1's	350
	Exercises for 13.5	350
13.6	Boolean Expressions	351
	Exercises for 13.6	355
13.7	A Brief Introduction to the Application of Boolean Algebra to Switching Theory	355
	Exercises for 13.7	361

CHAPTER 14	MONOIDS AND AUTOMATA	363
14.1	Monoids	364
	General Concepts and Properties of Monoids	365
	Monoid Isomorphisms	366
	Exercises for 14.1	366
14.2	Free Monoids and Languages	367
	Free Monoids Over an Alphabet	368
	Languages	368
	Two Fundamental Problems:	
	Recognition and Generation	369
	Phase Structure Grammars and Languages	371
	Exercises for 14.2	373
14.3	Automata, Finite-State Machines	374
	Recognition in Regular Languages	378
	Exercises for 14.3	379
14.4	The Monoid of a Finite-State Machine	380
	Exercises for 14.4	383
14.5	The Machine of a Monoid	384
	Exercises for 14.5	386
CHAPTER 15	GROUP THEORY AND APPLICATIONS	387
15.1	Cyclic Groups	388
	Application: Fast Adders	391
	Exercises for 15.1	394
15.2	Cosets and Factor Groups	394
	Exercises for 15.2	400
15.3	Permutation Groups	401
	Exercises for 15.3	409
15.4	Normal Subgroups and Group Homomorphisms	410
	Normal Subgroups	410
	Homomorphisms	413
	Exercises for 15.4	417
15.5	Coding Theory—Group Codes	417
	Binary Symmetric Channel	418