
Texts in Computer Science

Series editors

David Gries

Fred B. Schneider

More information about this series at <http://www.springer.com/series/3191>

Gerard O'Regan

Guide to Discrete Mathematics

An Accessible Introduction
to the History, Theory, Logic
and Applications

Gerard O'Regan
SQC Consulting
Mallow, Cork
Ireland

ISSN 1868-0941
Texts in Computer Science
ISBN 978-3-319-44560-1
DOI 10.1007/978-3-319-44561-8

ISSN 1868-095X (electronic)
ISBN 978-3-319-44561-8 (eBook)

Library of Congress Control Number: 2016948294

© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Lizbeth Román Padilla (Liz)
For sincere friendship

Preface

Overview

The objective of this book is to give the reader a flavor of discrete mathematics and its applications to the computing field. The goal is provide a broad and accessible guide to the fundamentals of discrete mathematics, and to show how it may be applied to various areas in computing such as cryptography, coding theory, formal methods, language theory, computability, artificial intelligence, theory of databases, and software reliability. The emphasis is on both theory and applications, rather than on the study of mathematics for its own sake.

There are many existing books on discrete mathematics, and while many of these provide more in-depth coverage on selected topics, this book is different in that it aims to provide a broad and accessible guide to the reader, and to show the rich applications of discrete mathematics in a wide number of areas in the computing field.

Each chapter of this book could potentially be a book in its own right, and so there are limits to the depth of coverage for each chapter. However, the author hopes that this book will motivate and stimulate the reader, and encourage further study of the more advanced texts.

Organization and Features

The first chapter discusses the contributions made by early civilizations to computing. This includes works done by the Babylonians, Egyptians, and Greeks. The Egyptians applied mathematics to solving practical problems such as the construction of pyramids. The Greeks made major contributions to mathematics and geometry.

Chapter 2 provides an introduction to fundamental building blocks in discrete mathematics including sets, relations and functions. A set is a collection of well-defined objects and it may be finite or infinite. A relation between two sets A and B indicates a relationship between members of the two sets, and is a subset of the Cartesian product of the two sets. A function is a special type of relation such

that for each element in A there is at most one element in the co-domain B . Functions may be partial or total and injective, surjective, or bijective.

Chapter 3 presents the fundamentals of number theory, and discusses prime number theory and the greatest common divisor and the least common multiple of two numbers. We also discuss the representation of numbers on a computer.

Chapter 4 discusses mathematical induction and recursion. Induction is a common proof technique in mathematics, and there are two parts to a proof by induction (the base case and the inductive step). We discuss strong and weak induction, and we discuss how recursion is used to define sets, sequences, and functions. This leads us to structural induction, which is used to prove properties of recursively defined structures.

Chapter 5 discusses sequences and series, and permutations and combinations. Arithmetic and geometric sequences and series and applications of geometric sequences and series to the calculation of compound interest and annuities are discussed.

Chapter 6 discusses algebra and simple and simultaneous equations, including the method of elimination and the method of substitution to solve simultaneous equations. We show how quadratic equations may be solved by factorization, completing the square or using the quadratic formula. We present the laws of logarithms and indices. We discuss various structures in abstract algebra, including monoids, groups, rings, integral domains, fields, and vector spaces.

Chapter 7 discusses automata theory, including finite-state machines, pushdown automata, and Turing machines. Finite-state machines are abstract machines that are in only one state at a time, and the input symbol causes a transition from the current state to the next state. Pushdown automata have greater computational power than finite-state machines, and they contain extra memory in the form of a stack from which symbols may be pushed or popped. The Turing machine is the most powerful model for computation, and this theoretical machine is equivalent to an actual computer in the sense that it can compute exactly the same set of functions.

Chapter 8 discusses matrices including 2×2 and general $m \times n$ matrices. Various operations such as the addition and multiplication of matrices are considered, and the determinant and the inverse of a matrix are discussed. The application of matrices to solving a set of linear equations using Gaussian elimination is considered.

Chapter 9 discusses graph theory where a graph $G = (V, E)$ consists of vertices and edges. It is a practical branch of mathematics that deals with the arrangements of vertices and edges between them, and it has been applied to practical problems such as the modeling of computer networks, determining the shortest driving route between two cities, and the traveling salesman problem.

Chapter 10 discusses cryptography, which is an important application of number theory. The code breaking work done at Bletchley Park in England during the Second World War is discussed, and the fundamentals of cryptography, including private and public key cryptosystems, are discussed.

Chapter 11 presents coding theory and concerns error detection and error correction codes. The underlying mathematics of coding theory is abstract algebra, and this includes group theory, ring theory, fields, and vector spaces.

Chapter 12 discusses language theory and grammars, parse trees, and derivations from a grammar. The important area of programming language semantics is discussed, including axiomatic, denotational, and operational semantics.

Chapter 13 discusses computability and decidability. The Church–Turing thesis states that anything that is computable is computable by a Turing machine. Church and Turing showed that mathematics is not decidable, in that there is no mechanical procedure (i.e., algorithm) to determine whether an arbitrary mathematical proposition is true or false, and so the only way is to determine the truth or falsity of a statement is by trying to solve the problem.

Chapter 14 presents a short history of logic and Greek contributions to syllogistic logic, stoic logic, fallacies, and paradoxes. Boole’s symbolic logic and its application to digital computing, and Frege’s work on predicate logic are discussed.

Chapter 15 provides an introduction to propositional and predicate logic. Propositional logic may be used to encode simple arguments that are expressed in natural language, and to determine their validity. The nature of mathematical proof along with proof by truth tables, semantic tableaux, and natural deduction is discussed. Predicate logic allows complex facts about the world to be represented, and new facts may be determined via deductive reasoning. Predicate calculus includes predicates, variables, and quantifiers, and a predicate is a characteristic or property that the subject of a statement can have.

Chapter 16 presents some advanced topics in logic including fuzzy logic, temporal logic, intuitionistic logic, undefined values, theorem provers, and the applications of logic to AI. Fuzzy logic is an extension of classical logic that acts as a mathematical model for vagueness. Temporal logic is concerned with the expression of properties that have time dependencies, and it allows temporal properties about the past, present, and future to be expressed. Intuitionism was a controversial theory on the foundations of mathematics based on a rejection of the law of the excluded middle, and an insistence on constructive existence. We discuss three approaches to deal with undefined values, including the logic of partial functions; Dijkstra’s approach with his *cand* and *cor* operators; and Parnas’ approach which preserves a classical two-valued logic.

Chapter 17 provides an introduction to the important field of software engineering. The birth of the discipline was at the Garmisch conference in Germany in the late 1960s. The extent to which mathematics should be employed in software engineering is discussed, and this remains a topic of active debate.

Chapter 18 discusses formal methods, which consist of a set of mathematic techniques that provide an extra level of confidence in the correctness of the software. They may be employed to formally state the requirements of the proposed system, and to derive a program from its mathematical specification. They may be employed to provide a rigorous proof that the implemented program satisfies its specification. They have been mainly applied to the safety critical field.

Chapter 19 presents the Z specification language, which is one of the most widely used formal methods. It was developed at Oxford University in the U.K.

Chapter 20 discusses probability and statistics and includes a discussion on discrete random variables; probability distributions; sample spaces; sampling; the abuse of statistics; variance and standard deviation; and hypothesis testing. The applications of probability to the software reliability field and queuing theory are briefly discussed.

Audience

The audience of this book includes computer science students who wish to gain a broad and accessible overview of discrete mathematics and its applications to the computing field. The book will also be of interest to students of mathematics who are curious as to how discrete mathematics is applied to the computing field. The book will also be of interest to the motivated general reader.

Acknowledgments

I am deeply indebted to family and friends who supported my efforts in this endeavor. I would like to thank Lizbeth Román Padilla (Liz) for sincere friendship over the years, and I wish her continued success with her Bayesian statistics. I would like to thank the team at Springer, and especially Wayne Wheeler and Simon Rees.

Cork, Ireland

Gerard O'Regan

Contents

1	Mathematics in Civilization	1
1.1	Introduction	1
1.2	The Babylonians	4
1.3	The Egyptians	6
1.4	The Greeks	8
1.5	The Romans	17
1.6	Islamic Influence	19
1.7	Chinese and Indian Mathematics	22
1.8	Review Questions	23
1.9	Summary	23
	References	24
2	Sets, Relations and Functions	25
2.1	Introduction	25
2.2	Set Theory	26
2.2.1	Set Theoretical Operations	28
2.2.2	Properties of Set Theoretical Operations	31
2.2.3	Russell's Paradox	32
2.2.4	Computer Representation of Sets	33
2.3	Relations	34
2.3.1	Reflexive, Symmetric and Transitive Relations	35
2.3.2	Composition of Relations	37
2.3.3	Binary Relations	39
2.3.4	Applications of Relations	40
2.4	Functions	41
2.5	Application of Functions	46
2.6	Review Questions	49
2.7	Summary	50
	References	51
3	Number Theory	53
3.1	Introduction	53
3.2	Elementary Number Theory	55
3.3	Prime Number Theory	59

3.3.1	Greatest Common Divisors (GCD)	61
3.3.2	Least Common Multiple (LCM)	62
3.3.3	Euclid's Algorithm	63
3.3.4	Distribution of Primes	65
3.4	Theory of Congruences.	67
3.5	Binary System and Computer Representation of Numbers	71
3.6	Review Questions	73
3.7	Summary	74
	References.	74
4	Mathematical Induction and Recursion	75
4.1	Introduction	75
4.2	Strong Induction	78
4.3	Recursion	80
4.4	Structural Induction.	82
4.5	Review Questions	83
4.6	Summary	83
	Reference	84
5	Sequences, Series and Permutations and Combinations	85
5.1	Introduction	85
5.2	Sequences and Series	86
5.3	Arithmetic and Geometric Sequences	87
5.4	Arithmetic and Geometric Series	88
5.5	Simple and Compound Interest.	89
5.6	Time Value of Money and Annuities	91
5.7	Permutations and Combinations	92
5.8	Review Questions	96
5.9	Summary	97
6	Algebra	99
6.1	Introduction	99
6.2	Simple and Simultaneous Equations	100
6.3	Quadratic Equations	103
6.4	Indices and Logarithms.	106
6.5	Horner's Method for Polynomials.	108
6.6	Abstract Algebra.	109
6.6.1	Monoids and Groups	109
6.6.2	Rings	111
6.6.3	Fields.	112
6.6.4	Vector Spaces	113
6.7	Review Questions	115
6.8	Summary	116
	Reference	116

7	Automata Theory	117
7.1	Introduction	117
7.2	Finite-State Machines	118
7.3	Pushdown Automata	121
7.4	Turing Machines	123
7.5	Review Questions	125
7.6	Summary	125
	Reference	126
8	Matrix Theory	127
8.1	Introduction	127
8.2	Two $\times$ Two Matrices	129
8.3	Matrix Operations	131
8.4	Determinants	133
8.5	Eigen Vectors and Values	135
8.6	Gaussian Elimination	136
8.7	Review Questions	138
8.8	Summary	138
	Reference	139
9	Graph Theory	141
9.1	Introduction	141
9.2	Undirected Graphs	143
9.2.1	Hamiltonian Paths	147
9.3	Trees	148
9.3.1	Binary Trees	149
9.4	Graph Algorithms	150
9.5	Graph Colouring and Four-Colour Problem	150
9.6	Review Questions	152
9.7	Summary	152
	References	153
10	Cryptography	155
10.1	Introduction	155
10.2	Breaking the Enigma Codes	157
10.3	Cryptographic Systems	160
10.4	Symmetric Key Systems	161
10.5	Public Key Systems	165
10.5.1	RSA Public Key Cryptosystem	167
10.5.2	Digital Signatures	168
10.6	Review Questions	169
10.7	Summary	169
	References	170

11	Coding Theory	171
11.1	Introduction	171
11.2	Mathematical Foundations	172
11.3	Simple Channel Code	173
11.4	Block Codes	174
11.4.1	Error Detection and Correction	176
11.5	Linear Block Codes	177
11.5.1	Parity Check Matrix	179
11.5.2	Binary Hamming Code	180
11.5.3	Binary Parity-Check Code	181
11.6	Miscellaneous Codes in Use	182
11.7	Review Questions	182
11.8	Summary	183
	References	183
12	Language Theory and Semantics	185
12.1	Introduction	185
12.2	Alphabets and Words	186
12.3	Grammars	187
12.3.1	Backus Naur Form	189
12.3.2	Parse Trees and Derivations	191
12.4	Programming Language Semantics	192
12.4.1	Axiomatic Semantics	193
12.4.2	Operational Semantics	195
12.4.3	Denotational Semantics	196
12.5	Lambda Calculus	197
12.6	Lattices and Order	199
12.6.1	Partially Ordered Sets	199
12.6.2	Lattices	201
12.6.3	Complete Partial Orders	202
12.6.4	Recursion	203
12.7	Review Questions	205
12.8	Summary	205
	References	206
13	Computability and Decidability	207
13.1	Introduction	207
13.2	Logicism and Formalism	208
13.3	Decidability	210
13.4	Computability	212
13.5	Computational Complexity	216
13.6	Review Questions	216
13.7	Summary	217
	Reference	218

14	A Short History of Logic	219
14.1	Introduction	219
14.2	Syllogistic Logic	220
14.3	Paradoxes and Fallacies	222
14.4	Stoic Logic	223
14.5	Boole's Symbolic Logic	225
14.5.1	Switching Circuits and Boolean Algebra	227
14.6	Application of Symbolic Logic to Digital Computing	229
14.7	Frege	230
14.8	Review Questions	231
14.9	Summary	232
	References	233
15	Propositional and Predicate Logic	235
15.1	Introduction	235
15.2	Propositional Logic	236
15.2.1	Truth Tables	238
15.2.2	Properties of Propositional Calculus	240
15.2.3	Proof in Propositional Calculus	242
15.2.4	Semantic Tableaux in Propositional Logic	244
15.2.5	Natural Deduction	246
15.2.6	Sketch of Formalization of Propositional Calculus	248
15.2.7	Applications of Propositional Calculus	248
15.2.8	Limitations of Propositional Calculus	250
15.3	Predicate Calculus	250
15.3.1	Sketch of Formalization of Predicate Calculus	253
15.3.2	Interpretation and Valuation Functions	255
15.3.3	Properties of Predicate Calculus	256
15.3.4	Applications of Predicate Calculus	256
15.3.5	Semantic Tableaux in Predicate Calculus	257
15.4	Review Questions	259
15.5	Summary	260
	References	261
16	Advanced Topics in Logic	263
16.1	Introduction	263
16.2	Fuzzy Logic	264
16.3	Temporal Logic	265
16.4	Intuitionist Logic	267
16.5	Undefined Values	269
16.5.1	Logic of Partial Functions	269
16.5.2	Parnas Logic	271
16.5.3	Dijkstra and Undefinedness	272
16.6	Logic and AI	274
16.7	Theorem Provers for Logic	278

16.8	Review Questions	279
16.9	Summary	280
	References.	280
17	Software Engineering Mathematics	283
17.1	Introduction	283
17.2	What Is Software Engineering?.	285
17.3	Early Software Engineering Mathematics	290
17.4	Mathematics in Software Engineering.	293
17.5	Software Inspections and Testing	294
17.6	Process Maturity Models.	295
17.7	Review Questions.	296
17.8	Summary	296
	References.	297
18	Formal Methods	299
18.1	Introduction	299
18.2	Why Should We Use Formal Methods?	302
18.3	Applications of Formal Methods.	303
18.4	Tools for Formal Methods	303
18.5	Approaches to Formal Methods	304
	18.5.1 Model-Oriented Approach	304
	18.5.2 Axiomatic Approach	305
18.6	Proof and Formal Methods	306
18.7	The Future of Formal Methods.	307
18.8	The Vienna Development Method	307
18.9	VDM ⁺ , the Irish School of VDM.	309
18.10	The Z Specification Language.	310
18.11	The B Method.	311
18.12	Predicate Transformers and Weakest Preconditions.	312
18.13	The Process Calculi	312
18.14	The Parnas Way	313
18.15	Usability of Formal Methods	314
18.16	Review Questions.	316
18.17	Summary	317
	References.	317
19	Z Formal Specification Language.	319
19.1	Introduction	319
19.2	Sets.	322
19.3	Relations.	323
19.4	Functions	325
19.5	Sequences	326
19.6	Bags	327
19.7	Schemas and Schema Composition.	328
19.8	Reification and Decomposition	331

19.9	Proof in Z.	332
19.10	Review Questions.	333
19.11	Summary	333
	Reference	334
20	Probability, Statistics and Applications	335
20.1	Introduction	335
20.2	Probability Theory	336
	20.2.1 Laws of Probability	337
	20.2.2 Random Variables	338
20.3	Statistics	342
	20.3.1 Abuse of Statistics	342
	20.3.2 Statistical Sampling	342
	20.3.3 Averages in a Sample	344
	20.3.4 Variance and Standard Deviation.	344
	20.3.5 Bell-Shaped (Normal) Distribution	345
	20.3.6 Frequency Tables, Histograms and Pie Charts.	347
	20.3.7 Hypothesis Testing	348
20.4	Software Reliability	350
	20.4.1 Software Reliability and Defects	351
	20.4.2 Cleanroom Methodology	353
	20.4.3 Software Reliability Models.	353
20.5	Queuing Theory	356
20.6	Review Questions.	358
20.7	Summary	359
	References.	360
	Glossary	361
	Index	365

List of Figures

Figure 1.1	The Plimpton 322 Tablet	5
Figure 1.2	Geometric representation of $(a + b)^2 = (a^2 + 2ab + b^2)$	6
Figure 1.3	Egyptian numerals	7
Figure 1.4	Egyptian representation of a number	7
Figure 1.5	Egyptian representation of a fraction	8
Figure 1.6	Eratosthenes measurement of the circumference of the earth	12
Figure 1.7	Archimedes in thought by Fetti	14
Figure 1.8	Plato and Aristotle	15
Figure 1.9	Julius Caesar	17
Figure 1.10	Roman numbers	18
Figure 1.11	Caesar cipher	19
Figure 1.12	Mohammed Al-Khwarizmi	20
Figure 1.13	Al Azhar University, Cairo	21
Figure 2.1	Bertrand russell	32
Figure 2.2	Reflexive relation	35
Figure 2.3	Symmetric relation	36
Figure 2.4	Transitive relation	36
Figure 2.5	Partitions of A	37
Figure 2.6	Composition of relations	38
Figure 2.7	Edgar Codd	41
Figure 2.8	PART relation	42
Figure 2.9	Domain and range of a partial function	42
Figure 2.10	Injective and surjective functions	45
Figure 2.11	Bijective function (One to one and Onto)	45
Figure 3.1	Pierre de Fermat	54
Figure 3.2	Pythagorean triples	55
Figure 3.3	Square numbers	55
Figure 3.4	Rectangular numbers	56
Figure 3.5	Triangular numbers	56
Figure 3.6	Marin Mersenne	57
Figure 3.7	Primes between 1 and 50	60
Figure 3.8	Euclid of Alexandria	63
Figure 3.9	Leonard Euler	67

Figure 6.1	Graphical solution to simultaneous equations	103
Figure 6.2	Graphical solution to quadratic equation	106
Figure 7.1	Finite state machine.	119
Figure 7.2	Deterministic FSM	119
Figure 7.3	Non-deterministic finite state machine.	120
Figure 7.4	Components of pushdown automata	121
Figure 7.5	Transition in pushdown automata	122
Figure 7.6	Transition function for pushdown automata M	123
Figure 7.7	Turing machine	123
Figure 7.8	Transition on turing machine.	125
Figure 8.1	Example of a 4×4 square matrix	128
Figure 8.2	Multiplication of two matrices.	132
Figure 8.3	Identity matrix I_n	133
Figure 8.4	Transpose of a matrix	133
Figure 8.5	Determining the (i, j) minor of A	134
Figure 9.1	Königsberg seven bridges problem	142
Figure 9.2	Königsberg graph	142
Figure 9.3	Undirected graph.	143
Figure 9.4	Directed graph.	143
Figure 9.5	Adjacency matrix.	144
Figure 9.6	Incidence matrix	144
Figure 9.7	Travelling salesman problem.	147
Figure 9.8	Binary tree.	149
Figure 9.9	Determining the chromatic colour of G	151
Figure 9.10	Chromatic colouring of G	151
Figure 10.1	Caesar Cipher	156
Figure 10.2	The Enigma machine.	157
Figure 10.3	Bletchley park	158
Figure 10.4	Alan Turing.	158
Figure 10.5	Replica of bombe	159
Figure 10.6	Symmetric key cryptosystem.	161
Figure 10.7	Public key cryptosystem	166
Figure 11.1	Basic digital communication	172
Figure 11.2	Encoding and decoding of an (n, k) block.	175
Figure 11.3	Error correcting capability sphere	177
Figure 11.4	Generator matrix	178
Figure 11.5	Generation of codewords.	179
Figure 11.6	Identity matrix $(k \times k)$	179
Figure 11.7	Hamming code $B(7, 4, 3)$ generator matrix	180
Figure 12.1	Noah chomsky. Courtesy of Duncan rawlinson.	188
Figure 12.2	Parse tree $5 \times 3 + 1$	192
Figure 12.3	Parse tree $5 \times 3 + 1$	192
Figure 12.4	Denotational semantics	197
Figure 12.5	Pictorial representation of a partial order.	199

Figure 12.6	Pictorial representation of a complete lattice	202
Figure 13.1	David Hilbert.	209
Figure 13.2	Kurt Gödel	212
Figure 13.3	Alonzo Church	212
Figure 14.1	Zeno of Citium	224
Figure 14.2	George Boole	225
Figure 14.3	Binary AND operation	228
Figure 14.4	Binary OR operation	228
Figure 14.5	NOT operation	228
Figure 14.6	Half-adder	229
Figure 14.7	Claude Shannon	230
Figure 14.8	Gottlob Frege	231
Figure 15.1	Gerhard gentzen	247
Figure 16.1	Conjunction and disjunction operators.	270
Figure 16.2	Implication and equivalence operators.	270
Figure 16.3	Negation	270
Figure 16.4	Finding Index in Array	272
Figure 16.5	Edsger Dijkstra. Courtesy of Brian Randell.	272
Figure 16.6	John McCarthy. Courtesy of John McCarthy.	275
Figure 17.1	David Parnas	286
Figure 17.2	Waterfall lifecycle model (V-model)	288
Figure 17.3	Spiral lifecycle model	288
Figure 17.4	Standish group report estimation accuracy.	289
Figure 17.5	Robert Floyd	290
Figure 17.6	Branch assertions in flowcharts	291
Figure 17.7	Assignment assertions in flowcharts	291
Figure 17.8	C.A.R Hoare	292
Figure 17.9	Watts Humphrey. Courtesy of Watts Humphrey	295
Figure 19.1	Specification of positive square root	320
Figure 19.2	Specification of a library system	321
Figure 19.3	Specification of borrow operation	321
Figure 19.4	Specification of vending machine using bags	328
Figure 19.5	Schema inclusion.	329
Figure 19.6	Merging schemas ($S_1 \vee S_2$).	329
Figure 19.7	Schema composition	331
Figure 19.8	Refinement commuting diagram	332
Figure 20.1	Carl Friedrich Gauss	345
Figure 20.2	Standard normal bell curve (Gaussian distribution).	346
Figure 20.3	Histogram test results	348
Figure 20.4	Pie chart test results.	348
Figure 20.5	Basic queuing system	356
Figure 20.6	Sample random variables in queuing theory	357