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

Springer

Gerard O'Regan
SQC Consulting
Mallow, Cork
Ireland

ISSN 1868-0941 ISSN 1868-095X (electronic)
Texts in Computer Science
ISBN 978-3-319-44560-1 ISBN 978-3-319-44561-8 (eBook)
DOI 10.1007/978-3-319-44561-8

Library of Congress Control Number: 2016948294

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

List of Figures