DISCRETE MATHEMATICS FOR ENGINEERS

English Edition, Revised and Enlarged

O. P. Kuznetsov

G. M. Adel'son-Vel'skii

DISCRETE MATHEMATICS FOR ENGINEERS

English Edition, Revised and Enlarged

O. P. Kuznetsov
Institute of Control Problems
USSR Academy of Sciences
Moscow, USSR

G. M. Adel'son-Vel'skii
Institute of Systems Research
USSR Academy of Sciences
Moscow, USSR

Translated from the Russian by
V. I. Kisin
Moscow, USSR

GORDON AND BREACH SCIENCE PUBLISHERS
NEW YORK LONDON PARIS MONTREUX TOKYO

Gordon and Breach Science Publishers

P.O. Box 786
Cooper Station
New York, NY 10276
United States of America

P.O. Box 197
London WC2E 9PX
England

58, rue Lhomond
75005 Paris
France

P.O. Box 161
1820 Montreux 2
Switzerland

14-9 Okubo 3-chome,
Shinjuku-ku,
Tokyo 160
Japan

Originally published in Russian as
Дискретная математика для инженера
by Izdatel'stvo Energiya, Moscow, USSR, 1980

Library of Congress Cataloging in Publication Data

Kuznetsov, O.P. (Oleg Petrovich)
Discrete mathematics for engineers.

Translation of: Diskretnaîa matematika dlia inzhenera.
1. Electronic data processing – Mathematics.
I. Adel'son-Vel'skiĭ, G.M. (Georgiĭ Maksimovich)
II. Title.
QA76.9.M35K8913 1985 519.4 84-13791

ISBN 2-88124-201-4. Printed in Great Britain by Bell and Bain Ltd., Glasgow.

DISCRETE MATHEMATICS FOR ENGINEERS

TOPICS IN COMPUTER MATHEMATICS

A series edited by David J. Evans, Loughborough University of Technology

Volume 1 PRECONDITIONING METHODS: ANALYSIS AND APPLICATIONS
Edited by David J. Evans

Volume 2 DISCRETE MATHEMATICS FOR ENGINEERS
By O. P. Kuznetsov and G. M. Adel'son-Vel'skii

Additional volumes in preparation

ISSN: 0275-5815

This book is part of a series. The publisher will accept continuation orders which may be cancelled at any time and which provide for automatic billing and shipping of each title in the series upon publication. Please write for details.

Introduction to the Series

In recent times there has been a broadly based and vigorous pursuit of mathematics and the need for it to interact with other sciences and disciplines, particularly the computer, is very great.

The rapid development and widespread use of computers in the academic, scientific, and industrial fields has stimulated the need to move away from the traditional areas of mathematics to a more computer-orientated approach to the subject.

For some time now, the emergence of a new form of mathematics has been evident and can most clearly be seen in the research papers submitted for publication to academic journals such as the *International Journal of Computer Mathematics.* This has led not only to an inevitable increase in its frequency of publication but also has prompted the publishers to introduce a new series of concise monographs, each concerned with one particular aspect of computer mathematics. There is a need to acquaint readers with the advances in these subjects, which are rapidly achieving tremendous importance and which are likely to impinge upon many aspects of our daily lives. Future volumes are planned at irregular intervals as dictated by need or the development of new topics.

David J. Evans

Preface to the English Edition

The English edition of *Diskretnaya Matematika dlya Inzhenera* has been enlarged considerably. A section on the well-known four-color conjecture has been added to Chapter 4, which outlines the recently obtained computer-generated solution. Chapter 6 has been augmented by a section on formal languages and grammars. The inclusion of this section in the book was stimulated by the extensive use of the notions and methods of grammar theory in systems programming, pattern recognition, and other fields of application. The exposition follows the main principle of the book: We do not present the applications of mathematics, but rather mathe*matics that has applications.* For this reason, virtually nothing has been said about the main application of grammars, i.e., methods of syntax analysis, although all the necessary concepts are introduced. In addition to familiar concepts and results, this section analyzes a new metalanguage for describing languages and grammars: networks of languages.

The theorem on the relationship between type-3 grammars and finite automata has been added to Chapter 7.

Chapter 8, which has also been revised and enlarged, describes and compares two independently evolved approaches to the classification of combinatorial problems by their complexity: (i) the NP-approach (which is well known in the English literature), based on nondeterministic Turing machines and the notion of NP-completeness, and (ii) the approach developed by Soviet mathematicians based on the concept of polynomial testing. A discussion of these approaches is preceded by a comparison of the complexity of the same problem with respect to different types of abstract machines. The definitions and proofs in this chapter are rather complex because they are necessarily tied to the programming of abstract machines, and programming always involves many tiresome unexciting details. This chapter therefore requires more hard work than the others.

The new list of references reflects these additions to the text. Furthermore, some monographs in Russian presenting standard material have been replaced with suitable counterparts in English. In contrast to principles guiding the compilation of reference lists in special monographs, the list given in this book does not outline the history of the subject, nor does it faithfully arrange priorities or offer a complete bibliographical description. Our purpose was to cite monographs and papers that present more detailed analyses of topics briefly discussed in this book.

Many terminological difficulties were encountered in translating the text to English. This is because the terminology of discrete mathematics has not yet been standardized, especially in graph theory. The task of reconciling, or at least describing, the discrepancies between terms is highly complicated. Our aim in this respect was to maintain a uniform terminological pattern throughout the book.

The authors are grateful to the translator, V.I. Kisin, for helping to overcome this difficulty in the English text and for eliminating many minor errors that had crept into the Russian edition.

Preface to the Russian Edition

This book was written for engineers, although the table of contents does not reflect this fact. Strictly speaking, the mathematics presented in the book is not applied mathematics, although we write for those who work in applied fields: It does not tell the reader where and how to apply the results or what the practical outcome will be. Nevertheless, this book is aimed at those who deal with technical subjects for the following reasons.

The fact that pure mathematics is the foundation of applied mathematics is a cliché. A book on electrical engineering does not give the definition of contour integral or vector product: It is taken for granted that the reader acquired this mathematical knowledge from a course in calculus. The inverse is equally true: A course in calculus for technical colleges, while purely mathematical, is oriented towards the applied problems likely to be encountered by graduates. However, such traditions have not yet developed in discrete mathematics. All application-oriented monographs usually start "from scratch," explaining what a disjunction or a loop in a graph is, while courses in pure discrete mathematics for engineers or engineering students, although they exist as academic curricula or lectures, are virtually nonexistent in the literature. One of the main purposes of this book is to develop such a course. Except for linear and discrete programming, this book covers almost all basic subjects that make up "nonprobabilistic" discrete mathematics.

We have also pursued another goal. Practical specialists often regard mathematics as a large handbook one must know how to use properly. Engineers are fond of formulas and methods but dislike theorems; they dislike even more the proofs of theorems (in monographs these proofs usually appear in smaller print). This is understandable. If the approach is application-oriented, knowing the proof does not provide any clues to the result: one needs to know "what" and "what for" and very seldom "how" and "why." This approach is nonetheless justifiable only in those fields in which time and effort have elaborated models of objects and processes have been established.

However, the main scientific problem in the field of control frequently involves the development of new models, rather than the use of existing models: Yesterday it was network diagrams and the logic of digital circuits, today it is the application of formal grammars to programming languages, tomorrow it will be fuzzy logic, etc. In this context, mathematics is used not as a method of calculation but as a way of thinking, as a language, as a means of formulating and organizing concepts. Such a command of mathematics requires a much higher educational level: an understanding of the importance of accurate statements and the ability to do without them when they interfere with getting to the heart of the problem, a certain "perception of the nontrivial", i.e., the ability to distinguish between a complicated problem and an impossible problem, to grasp the relationship between seemingly unrelated ideas and concepts.

We hope that this book will reduce the distance between the reader and this kind of mathematics. The book presents few techniques but many definitions and theorems. We tried to select them in such a way that they would be used as often as possible throughout this book, sometimes giving a different formulation of the same concepts in order to give a sense of unity and coherence to discrete mathematics. The same goal is pursued by the "jerky rhythm" of the presentation: Formal definitions and calculations are interspersed with informal arguments aimed to stimulate the reader to reflect upon the material presented.

The book is divided into four parts (not fully consistent with the order of chapters): (1) the language of discrete mathematics, (2) logic functions and automata, (3) the theory of algorithms and formal systems, and (4) graphs and discrete extremal problems.

The first part, devoted to sets and algebras (Chapters 1 and 2), is fairly traditional (the language of set theory is now taught in schools); in fact, it is a detailed glossary for the rest of the book. Its "linguistic" nature is betrayed, in particular, by the fact that it offers many definitions and examples but almost no theorems. The examples were selected to have the widest possible scope, in order to demonstrate the universality of the language of set theory. The reader familiar with this language may skip the first two chapters, returning to them later if necessary.

Logic and automata (Chapters 3 and 7) have also reached the stature of traditional elements in courses of discrete mathematics. In contrast to most of the textbooks that engineers use, this book treats these subjects as chapters of pure mathematics. The chapter on automata uses the concepts of the theory of algorithms and formal systems. The synthesis of circuits and automata is considered in much less detail for two reasons. First, most of the relevant problems are related to specific applied fields dealing with the design of digital circuits and thus lie outside the realm of pure mathematics; in addition, this topic has been discussed extensively in the literature. Second, the theory of circuits and automata, after having experienced an almost complete "biological cycle" during a single generation of researchers, offers an instructive example of vulnerability to moral obsolescence of technical fields that are weakly linked to fundamental theories. A decade ago all cybernetics journals contained many papers on minimization and synthesis. Most of them are now forgotten. In contrast, the ideas and results of the theory of automata, initially regarded as impracticable, such as set identification by automata, the Shannon–Yablonsky–Lupanov asymptotic theory of circuits, etc., have withstood the test of time. These results are now being used in various fields of cybernetics, such as computer science, complexity of calculations, artificial intelligence, etc.

The theories of algorithms and formal systems are the central themes of this book. These subjects are traditionally thought of as "ivory tower" science and are classified as subjects remote from practical problems and difficult to comprehend. The first of these prejudices is being slowly but irrevocably refuted by progress in theoretical cybernetics. In addition to the fact that this theory produced purely application-oriented branches related to algorithmic languages of programming and the complexity of computations, the awareness of the fundamental unsolvabilities in algorithm theory and the principles of organization of formal calculi have become essential elements of mathematical culture for all researchers dealing with algorithmization of control processes. This knowledge clarifies the capabilities and limitations of computers. Such understanding is particularly important today, since there are more computers than people who are able to use them efficiently. As for the prejudice of the difficulty of understanding (which arises largely because books on algorithm theory are written by mathematicians for the use of mathematicians), we can give the examples of two excellent books, by B.A. Trakhtenbrot and by M. Minsky (see the references to Chapters 5 and 6). We believe that a research engineer dealing with control and data processing problems, especially if he uses computers, can absorb the basic ideas of the theory of algorithms at least as well as a mathematician. In addition, an exposition of these ideas in a handbook gives mathematicians a rare opportunity to appeal to the programmer's intuition: for comprehension, this is at least as important as the store of conventional mathematical knowledge.

The sections dealing with graphs are included in Chapters 4 and 8. Chapter 4, like the first two chapters, is mostly linguistic. Most of the concepts discussed here are quite well known because graphs, due to their lucidity and universality, have very rapidly become the most common language of control problems. In contrast, Chapter 8 deals with highly specialized material contained in special publications. In addition to independent theoretical and application-oriented aspects of extremal problems and methods of solving them, we emphasize the general theoretical approach to these problems, primarily from the standpoint of algorithm theory. This chapter deals with relatively new problems that have gradually become very important in discrete mathematics, i.e., problems dealing with the complexity of computations. The results obtained in these fields are important because they are useful in evaluating the computer time and memory store required to solve specific problems. Like the general algorithm theory, which showed for the first time that there are unsolvable problems, its vigorously developing branch, the theory of complexity, gradually reveals to us that there are objectively complex problems (the so-called NP-complete problems are an example); the complexity can be absolute in a certain sense, i.e., it cannot be eliminated by increasing the capacity of the computers. An analysis of these problems is also instructive because almost all of them are combinations of a simple formulation and a complex solution.

Unfortunately, the test problems for each chapter had to be omitted because of the limitations of space. The shortcoming can be practically compensated for by carefully analyzing the numerous examples; we strongly recommend that the reader reconstruct the omitted parts of the proofs (this recommendation does not necessarily apply to the Church and Gödel theorems in Chapter 6).

One of our colleagues, A.Ya. Makarevskii, who has contributed significantly to the development of automata theory, encouraged the writing of this book. Sudden death at the age of thirty-seven prevented him from writing it himself. The authors dedicate this book to the memory of this extraordinary person and brilliant scientist.

O.P. Kuznetsov
G.M. Adel'son-Vel'skii

CONTENTS

CHAPTER ONE

Sets, Functions, and Relations

1.1. Sets and Operations with Sets

Sets and subsets. The concepts of a *set* and of *elements* of a set are the primary, fundamental concepts of mathematics. A set consists of its elements. A notation $a \in M$ denotes that element a belongs to the set M ("a belongs to M"); $a \notin M$ or $a \bar{\in} M$ denotes that a does not belong to the set M.

Example 1.1. M_1 is the set of all natural numbers: 0, 1, 2, 3 .... Below, we denote this set by N; the elements of N are natural numbers.

M_2 is the set of all natural numbers not exceeding 100.

M_3 is the set of all solutions of the equation $\sin x = 1$; the elements of M_3 are the numbers satisfying this equation.

M_4 is the set of all numbers of the type $\pi/2 \pm k\pi$, where $k \in N$.

M_5 is the set of all real numbers (denoted throughout the Section by R).

M_6 is a football team called "Bulls" (i.e., the set of all its players).

M_7 is the set of all major-league football teams in the 1982 season.

The elements of M_7 are football teams, i.e., sets of type M_6. Hence, sets may constitute elements of other sets; sets of sets (M_7), sets of sets of sets (the set of all professional football leagues), and so on, are possible.

Digression 1.1. As any other primary concept of a mathematical theory, the concept of a set is not defined. Indeed, any definition involves other concepts which logically precede the defined concept. As a result, at least the first definition of a theory necessarily contains undefined concepts which are regarded as primary concepts. Usually those concepts whose interpretation does not lead to appreciable controversy are chosen as primary concepts; more precisely, the concepts for which the differences in interpretation do not violate the correctness of the corollaries in the theory. This is precisely the concept of a set for the theories which are considered in this book. Further details on the basic principles of mathematical theories are given in Chapter 6. ∎

A set A is said to be a *subset* of a set B (denoted by $A \subseteq B$; symbol $\subseteq$ is called the inclusion symbol) if each element of A is an element of B. It is then said that B contains A. The sets A and B are equal if their elements coincide, i.e., if $A \subseteq B$ and $B \subseteq A$. The second version of defining the equality of sets points to the most frequent method of proving that the sets are equal: first the statement $A \subseteq B$ ("leftward") and then the statement $A \supseteq B$ ("rightward") are proved. Many mathematical theorems state that two sets are equal. One example is the trigonometric theorem $M_3 = M_4$ which consists of two statements: (i) each solution of the equation $\sin x = 1$ has the form $\pi/2 \pm k\pi$ ($M_3 \subseteq M_4$); (ii) any number of the type $\pi/2 \pm k\pi$ is a solution of the equation $\sin x = 1$ ($M_4 \subseteq M_3$).

If $A \subseteq B$ and $A \neq B$, then A is often called a proper subset of B (denoted by $A \subset B$; symbol $\subset$ is called the symbol of proper inclusion).

In dealing with sets of sets there is a risk of confusing the symbols $\in$ and $\subset$. For example, $M_6 \in M_7$ is true, but $M_6 \subset M_7$ is not (because the sets M_6 and M_7 consist of different elements!).

Sets may be finite or infinite (i.e., they may consist of a finite or an infinite number of elements. The number of elements in a finite set M is called the cardinality of M, which is often denoted by $|M|$. The cardinality of an infinite set is a more complex concept. It will be considered after the concept of correspondence is introduced.

A set with cardinality 0, i.e., one containing no elements, is called an empty set, or null set, and designated $\emptyset$ throughout. It is assumed that the null set is a subset of any set. The null set is introduced for the sake of convenience and universality of the mathematics language. For example, when one analyzes a set of objects possessing a specific property and then finds that no such objects exist, it is much more convenient to say that the set of interest is empty than to declare that it does not exist. The statement "the set M is not empty" is a more compact formulation of the equivalent statement "there are elements belonging to M."

How to define a set. A set can be defined by a list of its elements, by a generating procedure, or by a description of the properties characteristic of its elements.

Only a finite set can be defined by a list. A description of the type $N = 1, 2, 3 \ldots$ is not a list but a conditional notation admissible only if no misunderstanding is possible. A list is usually written in braces. For example, $A = \{a, b, d, h\}$ indicates that the set A consists of four elements a, b, d, and h.

A generating procedure describes the way to obtain the elements of a set from the elements already obtained or from other objects. All objects which can be constructed by such a procedure are regarded as elements of the set. One such example is the description of the set M_4, where the initial objects for the construction are the natural numbers, and the generating procedure is the calculation described by the formula $\pi/2 \pm k\pi$. Another example is the set $M_{2^n} = 1, 2, 4, 8, 16 \ldots$, whose generating procedure is defined by the following two rules: (i) $1 \in M_{2^n}$; (ii) if $m \in M_{2^n}$, then $2m \in M_{2^n}$. (Rules thus described are called *inductive* or *recursive*; we shall study them below.) The third example is the set M_π, which can be described as follows. Suppose we have a procedure for calculating the digits in the expansion of π into a non-terminating decimal fraction: $\pi = 3.1415926536\ldots$. As the calculation goes on, we shall form three-digit numbers from the successive digits of the expansion: 314, 159, 265, etc. The set of all these numbers will be denoted by M_π.

Another widely used generating procedure is the formation of sets from other sets, by means of operations with sets, which will be described below.

A typical way to represent a set is to describe the properties of its elements. In Example 1.1, this is done for the sets M_2, M_3, M_5; the description of M_4 can be interpreted as the description of one property of these elements, i.e., the possibility of writing them as $\pi/2 + k\pi$. The set M_{2^n} can be represented by a phrase "M_{2^n} is the set of all integers which are powers of two." If the relevant property of the elements of M can be described by a short expression P (x) (which is equivalent to saying that "x possesses a property P"), M is represented by writing $M = \{x | P(x)\}$, to be read as: "M is the set of x having the property P." For example, $M_{2^n} = \{x | x = 2^k$, where $k \in N\}$, and $M_4 = \{x | x = \pi/2 \pm k\pi$, where $k \in N\}$. It is natural to require that any description of the properties be exact and unambiguous. For example, different people will give different lists (maybe, empty lists) for the set of all good films of 1982; the criteria used for the selection will be different. Such a set cannot be considered as described unambiguously. A safe way to describe rigorously a property of the elements of a given set is to formulate a recognizing procedure (also said to be the decision procedure), which establishes for any object whether it possesses the given property and, hence, whether it belongs to a given set. For example, the decision procedure for M_{2^n}, i.e., for the property "to be a power of two," may be any method of factoring out an integer into prime factors.

Note that in this example the decision procedure is not a generating one. However, it can easily be turned into such a procedure: take successively all natural numbers and factor each one of them into prime factors; those numbers which do not include factors different from 2 are included into M_{2^n}. On the other hand, a generating procedure is not necessarily a decision procedure. In this connection, we suggest that the reader think over the set M_π, but warn him to beware of hasty conclusions. We shall return to this set later.

How to classify the representation of the set M_6? It cannot be classified: