国外优秀信息科学与技术系列教学用书

离散数学结构

——理论与应用

（影印版）

DISCRETE MATHEMATICAL STRUCTURES

Theory and Applications

■ D. S. Malik

高等教育出版社
Higher Education Press

国外优秀信息科学与技术系列教学用书

离散数学结构

——理论与应用

(影印版)

DISCRETE MATHEMATICAL STRUCTURES

Theory and Applications

D.S.Malik

高等教育出版社

图字:01 -2004 -5304 号

Discrete Mathematical Structures: Theory and Applications, First Edition
D. S. Malik
原版 ISBN:0 -619 -21285 -3

981 -254 -658 -8

图书在版编目(CIP)数据

离散数学结构:理论与应用 = Discrete Mathematical Structures: Theory and Applications/(美)马利克(Malik, D. S.)著.—影印本.—北京:高等教育出版社,2005.7
ISBN 7 -04 -016774 -3

Ⅰ.离... Ⅱ.马... Ⅲ.离散数学 - 英文
Ⅳ.O158

中国版本图书馆 CIP 数据核字(2005)第 067341 号

出版发行	高等教育出版社	**购书热线**	010 -58581118
社　址	北京市西城区德外大街 4 号	**免费咨询**	800 -810 -0598
邮政编码	100011	**网　址**	http://www.hep.edu.cn
总　机	010 -58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
			http://www.landraco.com.cn
经　销	北京蓝色畅想图书发行有限公司		
印　刷	北京民族印刷厂		
开　本	850 × 1168　1/16	**版　次**	2005 年 7 月第 1 版
印　张	41.25	**印　次**	2005 年 7 月第 1 次印刷
字　数	950 000	**定　价**	47.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

物料号　16774 -00

出版说明

20世纪末，以计算机和通信技术为代表的信息科学和技术对世界经济、科技、军事、教育和文化等产生了深刻的影响。信息科学技术的迅速普及和应用，带动了世界范围信息产业的蓬勃发展，为许多国家带来了丰厚的回报。

进入21世纪，尤其随着我国加入WTO，信息产业的国际竞争将更加激烈。我国信息产业虽然在20世纪末取得了迅猛发展，但与发达国家相比，甚至与印度、爱尔兰等国家相比，还有很大差距。国家信息化的发展速度和信息产业的国际竞争能力，最终都将取决于信息科学技术人才的质量和数量。引进国外信息科学和技术优秀教材，在有条件的学校推动开展英语授课或双语教学，是教育部为加快培养大批高质量的信息技术人才采取的一项重要举措。

为此，教育部要求由高等教育出版社首先开展信息科学和技术教材的引进试点工作。同时提出了两点要求，一是要高水平，二是要低价格。在高等教育出版社和信息科学技术引进教材专家组的努力下，经过比较短的时间，第一批由教育部高等教育司推荐的20多种引进教材已经陆续出版。这套教材出版后受到了广泛的好评，其中有不少是世界信息科学技术领域著名专家、教授的经典之作和反映信息科学技术最新进展的优秀作品，代表了目前世界信息科学技术教育的一流水平，而且价格也是最优惠的，与国内同类自编教材相当。这套教材基本覆盖了计算机科学与技术专业的课程体系，体同了权威性、系统性、先进性和经济性等特点。

目前，教育部正在全国35所高校推动示范性软件学院的建设，这也是加快培养信息科学技术人才的重要举措之一。为配合软件学院的教学工作，结合各软件学院的教学计划和课程设置，高等教育出版社近期聘请有关专家和软件学院的教师遴选推荐了一批相应的原版教学用书，正陆续组织出版，以方便各软件学院开展双语教学。

我们希望这些教学用书的引进出版，对于提高我国高等学校信息科学技术的教学水平，缩小与国际先进水平的差距，加快培养一大批具有国际竞争力的高质量信息技术人才，起到积极的推动作用。同时我们也欢迎教师和专家们对我们的教材引进工作提出宝贵的意见和建议。联系方式：hep.cs@263.net。

高等教育出版社

二〇〇二年九月

Preface

Discrete Mathematical Structures: Theory and Applications is an innovative text that introduces a new way of teaching the Discrete Structures course. A course in discrete structures is an integral part of the computer science curriculum. The class may consist of both math and computer science majors and can be taught either by the mathematics department or by the computer science department. Therefore, it is important that a course in discrete structures present a balance of theoretical concepts as well as their relevant applications.

Approach

The approach that we have taken in this book is a culmination of many years of experience. Our main objective is to make the learning of discrete mathematics easier and fun. Typically, in computer science, a course in discrete mathematics is taken just after programming courses. In many programs, this course becomes a prerequisite of other higher-level courses. In *Discrete Mathematical Structures: Theory and Applications*, we want to give students a solid foundation of theoretical concepts and their applications.

We have been teaching the discrete structures course for a number of years. The textbooks that we have come across tend to be either theory oriented or applications oriented. We do not believe in simply presenting the statement of a theorem and then showing its proof. Showing proof after proof is the surest way to discourage many students. On the other hand, showing application after application without the reinforcement of theoretical results is like following a cook book.

In *Discrete Mathematical Structures: Theory and Applications*, we show why theory is important and how theory connects with applications. Over the years, we have learned that giving an example before and after presenting a theoretical result makes learning easier and effective. Before writing a proof, we usually present examples to show the relevance of the concept. Moreover, we do not just show a proof, we show how the proof is constructed. The same methodology is followed when we present an algorithm. Before and/or after presenting an algorithm, we show how the algorithm works.

This book is written exclusively with students in mind. The language is user-friendly and conducive to learning. Very often we hear statements from students such as "How do I solve problems and write proofs?" To bridge this extremely important gap, we present a set of fully Worked-Out Exercises at the end of each section. These Worked-Out Exercises teach students how to solve problems as well as write proofs—they prepare students to do exercises on their own.

The book contains a rich collection of exercises. Furthermore, at the end of each chapter we include a set of Programming Exercises. Students are encouraged to solve these exercises in the programming language of their choice, such as Maple, C++, or Java.

Although this book is intended for a one-semester course, the book contains more material than could possibly be covered in this time frame. This gives the instructor flexibility in determining topic coverage. The book contains thirteen chapters, and they can be studied out of order depending on individual preference.

Organization and Coverage

Chapter 1 covers the basics of set theory, logic, and algorithms. We present the basic terminology used in set theory and various results used throughout the book. In the logic section, after presenting the basic material, such as statements and rules of inference, we show various proof techniques. Finally, in the algorithm section, we set the syntax used to write algorithms throughout the book.

In Chapters 2 and 3 we cover relations, posets, and closure of relations in in detail. We show how graphs and matrices are used to represent relations. Moreover, we use matrices to determine the transitive closure of relations on a finite set. Warshall's algorithm is covered in detail to find the transitive closure. We also show how relations are used in the design of relational databases.

Chapter 4 covers functions in detail. Other than covering various types of functions, we show the relationship between functions and strings.

Chapter 5 focuses on counting techniques. More specifically, we discuss basic counting principles—the addition and the multiplication principle, pigeonhole principles, permutations, combinations, binomial coefficients, and discrete probability. We also give various algorithms to generate permutations, combinations, and binomial coefficients.

Chapter 6 is concerned with advanced counting techniques using recurrence relations. Following this, we focus on solving linear homogenous recurrence relations and certain linear nonhomogenous recurrence relations. We are especially interested in linear nonhomogenous recurrence relations as they frequently appear in the analysis of algorithms that use divide and conquer techniques. We present enough results so that we can analyze the various algorithms.

Chapter 7 covers graphs in detail. Starting with basic graph theory definitions and terminology, we discuss topics such as subgraphs, walks, paths, circuits, isomorphism of graphs, planer graphs, and graph coloring. Also covered is a way to represent graphs in computer memory as well as various graph algorithms.

Chapter 8 focuses on trees, special types of trees, and determining spanning and minimal spanning trees. We close this chapter with a discussion of the transport network and present an algorithm to determine a maximal flow in a network.

Chapter 9 is concerned with Boolean algebra and its applications in the design of electric circuits.

The Web site accompanying this book contains the following additional material: applications of Boolean algebra in the design of switching circuits, characterization of regular languages by right congruences, nondeterministic finite automata with lambda transitions, and generating functions.

Syllabus Planning

Some of the ways the chapters can be studied are:

1. Study all the chapters in sequence.
2. Study the chapters in the sequence: 1, 2, 3, 4, 7, 8, 5, 6, 9.
3. Study the chapters in the sequence: 1, 2, 4, 3, 7, 8, 5, 6, 9.
4. Study the chapters in the sequence: 1, 2, 4, 3, 7, 8, 9, 5, 6.

Features

Every chapter in this book includes the following features. These features are both conducive to learning and allow students to learn the material at their own pace.

- *Learning Objectives* offer an outline of the concepts discussed in detail in the chapter.
- *Remarks* highlight important facts about the concepts introduced in the chapter.
- More than 390 visual diagrams, both extensive and exhaustive, illustrate difficult concepts.
- *Numbered Examples* illustrate the key concepts.
- *Worked-Out Exercises* is a set of more than 220 *fully* Worked-Out Exercises at the end of each chapter. These exercises teach how to solve problems and write proofs. We strongly recommend that students study these Worked-Out Exercises very carefully in order to learn problem-solving techniques.
- *Section Review* offers a summary of the concepts covered in the chapter.
- *Exercises* further reinforce learning and ensure that students have, in fact, learned the material.
- *Programming Exercises* challenge students to write programs with a specified outcome.

Teaching Tools

Discrete Mathematical Structures: Theory and Applications includes teaching tools to support instructors in the classroom. The ancillaries that accompany the textbook include an Instructor's Manual, Solutions, Test Banks, and Test Engine, PowerPoint presentations, and Figure Files. All teaching tools available with this book are provided to the instructor on a single CD-ROM and are also available on the Web at `www.course.com`.

Electronic Instructor's Manual. The Instructor's Manual that accompanies this textbook includes:

- Additional instructional material to assist in class preparation, including suggestions for lecture topics
- Solutions to all the exercises, including the Programming Exercises

ExamView® This objective-based test generator lets the instructor create paper, LAN, or Web-based tests from testbanks specifically designed for this Course Technology text. Instructors can use the QuickTest Wizard to create tests in fewer than five minutes by taking advantage of Course Technology's question banks—or create customized exams.

Solutions. The solution files for all programming exercises in C++ are available at `www.course.com`, and are also available on the Teaching Tools CD-ROM.

PowerPoint Presentations. Microsoft PowerPoint slides are included for each chapter. Instructors might use the slides in a variety of ways, including as teaching aids during classroom presentations or as printed handouts for classroom distribution. Instructors can add their own slides for additional topics introduced to the class.

Figure Files. Figure files allow instructors to create their own presentations using figures taken directly from the text.

Acknowledgements

There are many people that we must thank who, in one way or another, contributed to the success of this book. First, we would like to thank Dr. S.C. Cheng for his support and making suggestions to improve the text. We must also thank students who, during the preparation, were spontaneous in telling us if certain portions needed to be reworded for better understanding and clearer reading. We must thank Lee I. Fenicle, Director, Office of Technology Transfer, Creighton University, Dr. Randall L. Crist, and Dr. Ratish Basu Roy for their involvement, support, and for providing encouraging words when we needed them. We would like to acknowledge the feedback provided by Sunil Kumar Maity and Madhumita Mukherjee.

We owe a great deal to the following reviewers who patiently read each page of every chapter of the current version and made critical comments to improve on the book: Jim Ball, Indiana State University; Jose Cordova, University of Louisiana at Monroe; Joseph Klerlein, Western Carolina University; and Catherine Yan, Texas A&M University. The reviewers will recognize that their criticisms have not been overlooked and, in fact, made this a better book. Thanks to Development Editor Laurie Brown for carefully editing and promptly returning each chapter. All this would not have been possible without the planning of Managing Editor Jennifer Muroff and Product Manager Alyssa Pratt. Our sincere thanks to Jennifer Muroff and Alyssa Pratt, as well as to Aimee Poirier, Senior Production Editor, and also to the QA department of Course Technology for carefully testing the code. We would especially like to thank Burt LaFountain, QA Tester, for carefully reading the manuscript, solutions to all the exercises, and programming exercises. We would also like to thank Kate Deibel, University of Washington, for providing answers to the programming exercises in Chapters 1–8 and 11, and Jim Bishop, Bryant College, for his help with the biographies.

We are thankful to our parents for their blessings.

Finally, we are thankful for the support of our wives Sadhana and Monisha and our children Shelly, Nilanjan, Debanjan, and Shubhashree. They cheered us whenever we were overwhelmed during the writing of this book. We welcome any comments concerning the text. In spite of our diligent efforts there may still be room for improvement. Comments may be forwarded to the following e-mail address: `malik@creighton.edu` .

D.S. Malik

Contents

chapter 1

Foundations: Sets, Logic, and Algorithms

The objectives of this chapter are to:

- Learn about sets
- Explore various operations on sets
- Become familiar with Venn diagrams
- Learn how to represent sets in computer memory
- Learn about statements (propositions)
- Learn how to use logical connectives to combine statements
- Explore how to draw conclusion using various argument forms
- Become familiar with quantifiers and predicates
- Learn various proof techniques
- Explore what an algorithm is

This chapter sets the stage for all that follows and also serves as an appropriate place for codifying certain technical terminologies used throughout the text. In the first section, we discuss sets and their basic properties. We then study mathematical logic in some detail. In this book, the focus is not just on theory but also on applications. When theoretical concepts are presented, we give various examples to clarify the concepts as well as to prove theoretical results, wherever appropriate. Therefore, after discussing sets, we study mathematical logic and describe various proof techniques.

Over the years a revolution in computer technology has changed the ways in which we live and communicate. Computer programs have made tedious computations easy to handle and have enabled us to achieve results quickly and to a great degree of precision. Therefore, throughout the book we discuss various algorithms that can be implemented in a variety of programming languages, such as C++ and Java. In the last section of this chapter, we introduce algorithms and describe the syntax of the pseudocode used to describe algorithms in this book.

Natural numbers, integers, rational numbers, and real numbers are a great source of examples. We assume that the reader is familiar with these number systems.

1.1 SETS

The mathematical theory of sets grew out of the German mathematician Georg Cantor's study of trigonometric series and series of real numbers. The language of sets has since become an important tool for all branches of mathematics, serving as a basis for the precise description of higher concepts and for mathematical reasoning.

Let us begin with the question, what is a *set*? It is fascinating that the answer to this very basic and apparently simple question once jeopardized the very foundation of set theory. In this text, however, we adopt a naive and intuitive point of view and introduce the definition of a **set** according to his definition. According to his definition, *a set is a well-defined collection of distinct objects* of our perception or of our thoughts, to be conceived as a whole.

Historical Notes

Georg Cantor (1845–1918)

Although considered one of the great German mathematicians, Cantor was born in St. Petersburg, Russia, in the winter of 1845 to a wealthy Danish merchant. At the age of 11, he moved with his family to Germany where he continued his education, earning a doctorate degree from the University of Berlin in 1867. In 1869 Cantor accepted a post at the University of Halle, an undistinguished school for women. His provocative ideas regarding concepts of infinity had put him in bad standing with his contemporaries, and many of them opposed his appointment to the prestigious University of Berlin. Cantor suffered fits of depression due in large part to stress related to his work. He spent the better part of his later years in and out of mental hospitals and ultimately died in a sanatorium.

Cantor is considered to be the founder of set theory, and he established its relation to transfinite numbers. He explored paradoxes that had existed in mathematics for centuries and even stumbled upon one of his own, now known as Cantor's paradox. Although his theories were vehemently disputed by his peers, including Leopold Kronecker, his mentor at the University of Berlin, modern mathematicians completely accept Cantor's work.

To develop a perfectly balanced working idea of sets, it is sufficient for a beginner to concentrate on the first part (the italicized part) of this definition. Note that here *well-defined* is an adjective to the noun *collection* and not to the distinct objects that are to be collected to form a set. What this means is that there should be no ambiguity whatsoever regarding the membership of such a collection; *well-defined* means that we can tell for certain whether an object is a member of the collection or not. These objects are called *members* or *elements* of the set.

For example, we can talk about the set of all positive integers, even though no one really knows all of them. But a collection of *some* positive integers is *not* a set because it is not clear whether a particular positive integer, say 5, is a member of this collection or not. For another example, the collection of students taking the discrete mathematics course in your school is a set. On the other hand, the collection of best cars in a city cannot be a set because there is no well-defined notion of best.

We use italic uppercase letters, $A, B, C, \ldots, X, Y, Z$, to denote sets. A set can be described in various ways, but the main point of any description is to specify the elements of the set in some unambiguous way. One common way, called the **roster method**, to describe a set is to list the elements of the set and enclose them within curly braces. For example, if A is a set of vowels, then we write

$$A = \{a, e, i, o, u\}.$$

For another example, we can describe the set B of all positive integers less than 11 as

$$B = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}.$$

Let X be a set. If x is an element of X, then we write $x \in X$ and say that x belongs to X. The symbol $\in$ stands for *belongs to*, which, like many other notations, was introduced in 1889 by the Italian mathematician Giuseppe Peano (1858–1932) and is believed to be a stylized form of the Greek epsilon. If x is not an element of X, then we write $x \notin X$ and say that x is not an element of X. The symbol $\notin$ stands for *does not belong to.*

EXAMPLE 1.1.1 Let A be the set

$$A = \{1, 2, 3, 4, 5\}.$$

Then $2 \in A$ and $5 \in A$. Also, $6 \notin A$.

EXAMPLE 1.1.2 Let B be the set of first 10 positive odd integers. Then

$$B = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19\}.$$

It follows that $9 \in B$ and $2 \notin B$.

We also describe sets in the following manner. Let S be a set. The notation

$$A = \{x \mid x \in S, P(x)\}$$

or

$$A = \{x \in S \mid P(x)\}$$

means that A is the set of all elements x of S such that x satisfies the property P. This way of describing a set is called the **set-builder method**.

For example, if $\mathbb{Z}$ denotes the set of integers, then

$$\mathbb{N} = \{x \mid x \in \mathbb{Z},\ x > 0\}$$

or

$$\mathbb{N} = \{x \in \mathbb{Z} \mid x > 0\}.$$

Here the property $P(x)$ is

$$P(x) : x > 0.$$

EXAMPLE 1.1.3 In set-builder notation, the set B of Example 1.1.2 can be described as

$$B = \{x \in \mathbb{Z} \mid x \text{ is odd and } 1 \leq x \leq 19\}.$$

EXAMPLE 1.1.4 Let $A = \{2, -2\}$. Because 2 and -2 are the only integers that satisfy the equation $x^2 - 4 = 0$, we can also write A as

$$A = \{x \mid x \in \mathbb{Z},\ x^2 - 4 = 0\}$$

or

$$A = \{x \in \mathbb{Z} \mid x^2 - 4 = 0\}.$$

Here the property $P(x)$ is

$$P(x) : x^2 - 4 = 0.$$

EXAMPLE 1.1.5 Let A be the set described in set-builder form as:

$$A = \{x \mid x \text{ is a complex number and } x^4 = 1\}.$$

Now the equation $x^4 = 1$, i.e., $x^4 - 1 = 0$, can be factored as

$$(x + 1)(x - 1)(x - i)(x + i) = 0,$$

where $i^2 = -1$ or $i = \sqrt{-1}$. This implies that the solutions of the equation $x^4 = 1$, where x is a complex number, are $x = 1, -1, i, -i$. Therefore, using the roster form, the set A can be written as

$$A = \{1, -1, i, -i\}.$$

Throughout the book, we will use numbers to provide examples. Therefore, we would like to standardize the symbols to denote various sets of numbers as follows.

$\mathbb{N}$: The set of all natural numbers (i.e., all positive integers)

$\mathbb{Z}$: The set of all integers

$\mathbb{Z}^*$: The set of all nonzero integers

$\mathbb{E}$: The set of all even integers

$\mathbb{Q}$: The set of all rational numbers

$\mathbb{Q}^*$: The set of all nonzero rational numbers

$\mathbb{Q}^+$: The set of all positive rational numbers

$\mathbb{R}$: The set of all real numbers

$\mathbb{R}^*$: The set of all nonzero real numbers

$\mathbb{R}^+$: The set of all positive real numbers

$\mathbb{C}$: The set of all complex numbers

$\mathbb{C}^*$: The set of all nonzero complex numbers

We know that every integer is a real number; that is, every element of $\mathbb{Z}$ is an element of $\mathbb{R}$. Similarly, every vowel is a letter in the set of English letters. In other words, if $A = \{a, e, i, o, u\}$ and B is the set of all English letters, then every element of A is an element of B. When every element of a set, say A, is also an element of a set, say B, we say that A is a subset of B. More formally, we have the following definition.

DEFINITION 1.1.6 ▶ Let X and Y be sets. Then X is said to be a **subset** of Y, written $X \subseteq Y$, if every element of X is an element of Y. If X is not a subset of Y, then we write $X \nsubseteq Y$.

EXAMPLE 1.1.7

(i) Let $X = \{0, 2, 4, 6, 8\}$, $Y = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, and $Z = \{1, 2, 3, 4, 5\}$. Then $X \subseteq Y$ because every element of X is an element of Y. However, because $0 \in X$ and $0 \notin Z$, we have $X \nsubseteq Z$.

Notice that we used the fact that $0 \in X$ and $0 \notin Z$ to conclude that $X \nsubseteq Z$. We could have also used the fact that $6 \in X$ and $6 \notin Z$ or $8 \in X$ and $8 \notin Z$ to conclude that $X \nsubseteq Z$. In other words, the elements 6 and 8 also prevent X from being a subset of Z.

(ii) Let $A = \{a, b, c\}$ and $B = \{a, c, b\}$. Now every element of A is also an element of B and so $A \subseteq B$. Also notice that $B \subseteq A$.

(iii) Let $A = \{$Basic, Fortran, C++$\}$ and $B = \{$Basic, Fortran, Pascal, C++, Java$\}$. Then $A \subseteq B$.

Note: For every set X, we have $X \subseteq X$.

Let X and Y be sets. If $X \subseteq Y$, we also say that X is contained in Y, or Y contains X, or Y is a **superset** of X (written $Y \supseteq X$).

Notice that in Example 1.1.7(i), every element of X is an element of Y. However, there are some elements in Y that are not in X. Such a set X is called a proper subset of Y.

DEFINITION 1.1.8 ▶ Let X and Y be sets. Then X is a **proper subset** of Y, written $X \subset Y$, if X is a subset of Y and there exists at least one element in Y that is not in X.

EXAMPLE 1.1.9

Let $A = \{a, b\}$ and $B = \{a, b, c\}$. Because every element of A is an element of B, we have $A \subseteq B$. Now $c \in B$ and $c \notin A$. Therefore, there exists an element in B that is not in A. It now follows that A is a proper subset of B, i.e., $A \subset B$.

EXAMPLE 1.1.10

The set of all even integers is a proper subset of the set of all integers. In set notation, $\{2n \mid n \in \mathbb{Z}\} \subset \mathbb{Z}$.

Note that $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$.

DEFINITION 1.1.11 ▶ Two sets X and Y are said to be **equal**, written $X = Y$, if every element of X is an element of Y and every element of Y is an element of X, i.e., if $X \subseteq Y$ and $Y \subseteq X$.

EXAMPLE 1.1.12

(i) $\{a, b, c\} = \{a, c, b\}$.
(ii) Let $A = \{1, 2, 3, 4\}$ and $B = \{x \mid x \text{ is a positive integer and } x^2 < 18\}$. Then $A = B$.
(iii) The set $A = \{x \mid x \text{ is an integer and } x^3 = 1\}$ and $B = \{1\}$ are equal.

Now consider the set $A = \{x \in \mathbb{Z} \mid x^2 - 2 = 0\}$. Notice that $x^2 - 2 = (x - \sqrt{2})(x + \sqrt{2})$. Thus, the solutions of the equation $x^2 - 2 = 0$ are $\sqrt{2}$ and $-\sqrt{2}$ and none of these is an integer. Therefore, it follows that A does not contain any elements. This is an empty collection of objects. We call it an empty set.

DEFINITION 1.1.13 ▶ A set is said to be an **empty** (or **null**) **set** if it has no elements. We denote an empty set by the symbol $\emptyset$.

Note: We can consider the empty set a subset of every set. In fact, if $\emptyset \nsubseteq A$ for some set A, then there exists an element $x \in \emptyset$ such that $x \notin A$. However, there is no such element x because $\emptyset$ is empty. Hence, $\emptyset \subseteq A$ for every set A.

In Example 1.1.1, the set A has five elements.

DEFINITION 1.1.14 ▶ Let X be a set.

(i) If there exists a nonnegative integer n such that X has n elements, then X is called a **finite set** with n elements.
(ii) X is called an **infinite set**, if X is not a finite set.

EXAMPLE 1.1.15

(i) The set $A = \{a, b, c\}$ has three elements, so it is a finite set.
(ii) The set B of the first 10 positive odd integers is a finite set. Notice that

$$B = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19\}.$$

(iii) The set of positive integers is an infinite set.

REMARK 1.1.16 ▶ Note that an empty set is a finite set with 0 elements.

Let S be a finite set with n distinct elements, where $n \geq 0$. Then we write $|S| = n$ and say that the *cardinality* (or the *number of elements*) of S is n.

If $A = \{a, b, c, d, e\}$, then A is a finite set with five elements and so $|A| = 5$.

Let

$$B = \{x \mid x \text{ is a positive even prime integer}\}.$$

The only positive even prime integer is 2. Therefore, $B = \{2\}$ and so $|B| = 1$.

A set consisting of only one element is called a **singleton set** or, simply, a **singleton**. Thus, B is a singleton.

Sometimes an infinite set is described using the roster method. For example, if $\mathbb{N}$ denotes the set of positive integers, then we can write $\mathbb{N} = \{1, 2, 3, \ldots\}$.

Let $X = \{1, 2\}$. Then $\emptyset \subseteq X$, $\{1\} \subseteq X$, $\{2\} \subseteq X$, and $X \subseteq X$. Now each of the sets $\emptyset$, $\{1\}$, $\{2\}$, X is well defined. We can therefore form the collection $\{\emptyset, \{1\}, \{2\}, X\}$ of these sets, which would itself be a set.

DEFINITION 1.1.17 ▶ For any set X, the **power set** of X, written $\mathcal{P}(X)$, is the set of all subsets of X. That is,

$$\mathcal{P}(X) = \{A \mid A \subseteq X\}.$$

For example, let $X = \{a, b, c\}$. Then

$$\mathcal{P}(X) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, X\}.$$

Notice that $|X| = 3$ and $|\mathcal{P}(X)| = 8 = 2^3$.

REMARK 1.1.18 ▶ Let X be a finite set X such that $|X| = k$. In Appendix B, we will show that $|\mathcal{P}(X)| = 2^k$.

REMARK 1.1.19 ▶ Because $\emptyset$ is a subset of every set, we have $\emptyset \subseteq \emptyset$. Therefore, we have $\mathcal{P}(\emptyset) = \{\emptyset\}$.

To avoid the logical difficulties that arise in the foundation of set theory, we further assume that each discussion involving a number of sets takes place with respect to an *arbitrarily chosen but fixed* set. This set is called a **universal set**[1] for that discussion and is generally denoted by U. All the sets under consideration in the problem must be subsets of U.

For example, in a discussion involving the sets $X = \{1, 2, 3\}$, $Y = \{2, 4, 6, 8\}$, $Z = \{1, 3, 5, 7\}$, one may choose $U = \{x \in \mathbb{N} \mid 1 \leq x \leq 8\}$ as a universal set. Moreover, any superset of U can also be considered a universal set for these sets X, Y, and Z.

Venn Diagrams

Typically, it is not easy to visualize a set. In 1880, however, the English logician John Venn devised a pictorial representation for sets and their fundamental operations. Though admittedly loose and imprecise, and therefore somewhat contrary to the spirit of logical rigor at the heart of set theory, one may still find this diagrammatic approach very convenient in developing the so-called *abstract visualization*, which is essential to *seeing* the mental image of these abstract happenings. In these

Historical Notes

John Venn (1834–1923)
John Venn was born to a well-to-do Anglican family in 1834. He studied at Caius College at Cambridge University, and in 1857 earned a B.A. degree. He was also chosen as a Fellow to the college, where he later lectured in the moral sciences in a position that he held for the remainder of his life. Also very important in Venn's life was religion; he was ordained first as deacon and then as priest, but given the climate of the times, he renounced his clerical orders in 1883.

Venn is best known for his diagrams that demonstrate sets and their unions, which he first introduced in a paper titled "On the Diagrammatic and Mechanical Representation of Propositions and Reasoning." Chiefly interested in the workings of logic, Venn wrote three books on the subject, in which he explored the mathematical logic of Boole as well as symbolic logic. Venn also wrote extensively on the history of Cambridge University, a work that is still being continued today.

[1] We want to make it very clear that in spite of what the name may seem to suggest, we are by no means proposing a set that is *universal* for all the problems. Rather, it may vary from problem to problem and even more: For a problem involving certain sets, the choice of a universal set is not unique, but once chosen, subject to the conditions stated above, it must be kept fixed throughout the subsequent discussions of that problem.